

Service Oriented Computing for Dynamic Virtual Learning Environments

Ajlan Suliman Al-Ajlan

A Dissertation Submitted in partail fulfilment
of the requirements for the degree of

Doctor of Philosophy

Software Technology Research Laboratory

De Montfort University, United Kingdom

2008



IN THE NAME OF ALLAH
THE MOST GRACIOUS, THE MOST MERCIFUL

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

This work is original work undertaken by me for the degree of Doctor of Philosophy, at De Montfort University, United Kingdom.

This thesis is written by me and produced using LATEX.

Dedication

I would like to dedicate this thesis to my mother (Helleh Mohammed Al-Ajlan) and my beloved wife (Sameyah Ali Al-Sallal) and to my children Suliman, Ria, and Abdulsalam.

Abstract

Using the Internet for teaching and learning has become a trend in modern higher education, facilitated through the exploitation of advanced computing technologies. Virtual Learning Environment (VLE) applications support online learning over the Internet, and VLEs have thus emerged as e-learning domains that are essential prerequisites in cutting edge design and implementation technologies in education.

Service Oriented Computing (SOC), as a novel software development and implementation approach, has become an active area of research and development. Web services, as an example of SOC, support the integration of software applications in an incremental way, using existing platforms and languages that utilize and adopt existing legacy systems. Thus, VLEs should be particularly well suited to Web services through the SOC approach. VLE services is a field subjected to continuous development but VLEs as Web services are still not generally accessible for academic institutions, although they have been adopted by some scientific projects. The next generation of VLEs should address the limitations of the current online systems by providing a richer context for online learning, one that is sensitive to the specific domain requirements of e-learning.

Web Services Matching and Selection (WSMS), as a part of the functional requirements of Web services, has received less attention from SOC researchers. It involves discovering a set of semantically equivalent services by filtering a set of available services based on service metadata, and instantaneously selecting the best

possible service. WSMS is the discovery of a service by a user, where correspondence is established between the objectives of the consumer and the capabilities of the service. It thereby aims to match and select the optimal service that best meets the requestor's needs.

The main aim of this doctoral work is to explore novel architectural designs for VLEs, based on the SOC paradigm and its related techniques. In addition, this investigation aims to extend the core ideas behind VLE tools, which are gradually becoming dominant within academic institutes. Another aim is to devise a policy-based technique to enforce security requirements for VLEs and to build a test-bed for VLE security based on Modular Moodle.

The fundamental contribution of this thesis is that it demonstrates that VLEs can be considered as services, which can be published, discovered and composed as perceived in the SOC paradigm. An additional contribution to the knowledge is that it has built a new extension to the structure of Web services: the Web Services Matching and Selection (WSMS) system. Another contribution to the knowledge is that traditional security requirements have been modified to cater for the highly mobile and changeable environment of VLEs; this has been achieved through policy-based techniques. These contributions to the body of knowledge have been published in learned journals and at conferences.

Acknowledgments

In the name of Allah, the Most Merciful and the Most Gracious, I give praise and thanks to Him for supporting me with the strength to complete this thesis, and for providing me with knowledgeable and caring individuals during the study process. This thesis could not have been completed without the recommendations, advice and suggestions of many people. It may not be possible to mention all of them here, but their contributions, guidance and support are extremely appreciated.

I would like to thank both Dr. Amelia Platt, who was my first supervisor, and Mr. Ralph Birkenhead who was my second supervisor, both of whom have encouraged my autonomy and given me invaluable instruction. In addition, I am truly grateful to Prof. Hussein Zedan who was my personal advisor; he was generous with me, provided freely of his knowledge and made many suggestions. His devotion and commitment to excellence provided me with the motivation, inspiration and encouragement that I needed to remain focused. Because of him, I am confident that I will make meaningful contributions to research.

I also wish to acknowledge contributions from many others, especially Prof. Martin Dougiamas who is the author of Moodle, and also Dr. Justin Filip and Dr. Monika Solanki. I also acknowledge those individually with whom we discussed issues that are addressed in this thesis, including Dr. Ahmad Al-Gamdi, Dr. Abdul-samad Al-Marghilani, Dr. Osamah Al-Hassan, Dr. Bassam Zafar, and Dr. Ahmad Al-Zahrani.

There are additional people, too numerous to mention by name, who have made a contribution to my knowledge and made the research task more gratifying. They include the people who made comments and suggestions on my research presentation, the survey participants for their cooperation, encouragement and support, the staff and officials at the School of Computer Sciences and Engineering at De Montfort University, and especially those at the Software Technology Research Laboratory (STRL). I am indebted to Qassim University for its encouragement and financial support, and to the Saudi Arabian Cultural Bureau in London, for its support and encouragement.

My final and greatest debt is to my family - my dear mother, my wife, my children (Suliman, Ria and Abdulsalam), my brothers and my sisters. The list of sacrifices they have made in order for me to complete this study is endless. Their support and love have been wonderful.

Contents

Declaration	ii
Abstract	iv
Acknowledgments	vi
1 Introduction	1
1.1 Background	1
1.2 Motivation of the Research	2
1.3 Aim & Objectives of the Research	3
1.4 Research Questions	5
1.5 Research Methodology and Evaluation	5
1.5.1 Methodology	6
1.5.2 Evaluation and Testing	8
1.6 Contributions of the Thesis	9
1.7 Thesis Structure and Reading Guide	11
2 Service-Oriented Computing	15
2.1 Introduction	15
2.2 Service-Oriented Computing	16
2.2.1 Definition of Service-Oriented Computing	17
2.2.2 Benefits of Service-Oriented Computing	18

2.3	Web Services	19
2.3.1	Definition of Web Services	19
2.3.2	The Benefits of Web Services	20
2.3.3	Web Services Architecture	22
2.4	Legacy Systems	25
2.5	The Web Services Technology Stack	26
2.5.1	Extensible Markup Language	26
2.5.2	Service Transport	28
2.5.3	XML Messaging	28
2.5.4	Service Description	33
2.5.5	Service Discovery	39
2.6	What are Web Services Good for?	40
2.7	Who is Using Web Services?	40
2.8	The Future of Web Services	41
2.9	Summary	43
3	Virtual Learning Environments as E-learning Systems	44
3.1	Introduction	44
3.2	Virtual Learning Environments	45
3.2.1	The Definition of Virtual Learning Environments	46
3.2.2	The History of Virtual Learning Environments	48
3.2.3	The Benefits of Virtual Learning Environments	48
3.2.4	The Architecture of Virtual Learning Environments	51
3.2.5	Functional and Non-Functional Requirements of Virtual Learning Environments	53
3.2.6	The Main Aspects of Evaluation for Virtual Learning Environments	54
3.2.7	The Challenge of Virtual Learning Environments	55

3.3	The Main Features and Tools of Virtual Learning Environments . . .	57
3.4	Open Source Softwares	58
3.5	The Future of Virtual Learning Environments	63
3.6	Summary	65
4	A Comparison of Current VLE Features	66
4.1	Introduction	66
4.2	Comparative Study of VLE Products	68
4.2.1	Comparative Study Based on Features and Capabilities of VLE Tools	68
4.2.2	Comparison Based on Focusing on the Technical Aspects of the VLE Systems	75
4.3	Moodle as a Selected Platform	79
4.3.1	The Definition of Moodle	79
4.3.2	The History of Moodle	80
4.3.3	The Reasons for Choosing Moodle	81
4.3.4	The Limitations of Moodle	84
4.3.5	Architecture of Moodle Sites	85
4.4	Virtual Learning Environment Tools of Moodle	89
4.4.1	Activity Modules	89
4.4.2	Resource Types	91
4.5	Assignments Module	92
4.5.1	Assignment Types	93
4.6	Websites are using Moodle all over the World	96
4.6.1	Moodle in the Kingdom of Saudi Arabia	97
4.7	Summary	99
5	Production of Technical Requirements for SOC Based on VLE (Moodle)	100

5.1	Introduction	100
5.2	Distributed System	101
5.3	The Distributed System of Virtual Learning Environments	104
5.4	An Example: Web Services for Moodle	105
5.5	Requirements for Virtual Learning Environments (Moodle) Based on Web Services	108
5.5.1	Functional Requirements	108
5.5.2	Non-Functional Requirements	114
5.6	Summary	121
6	The Architecture for an SOC-Based VLE (Moodle)	122
6.1	Introduction	122
6.2	Unified Modeling Language	123
6.2.1	History of Unified Modelling Language	124
6.2.2	The Aims of UML	125
6.2.3	Why Use UML?	125
6.3	The Architecture for an SOC-based VLE (Moodle)	127
6.4	Matching for Specific Type	136
6.5	The Architecture for WSMS with two VLE Platforms (Moodle) . . .	139
6.6	Finite State Machine	144
6.7	Summary	149
7	Devise A Policy-Based Techniques for Enforce Security Environ- ments of VLE	150
7.1	Introduction	150
7.2	A Comparative Study between Policy Specification Languages	152
7.3	Ponder Policies	155
7.4	Types of Ponder Policy	157
7.4.1	Access Control Policies	157

7.4.2	Obligation policies	161
7.5	Graph Representation of Ponder Policy	162
7.5.1	Domain	162
7.5.2	Operation	163
7.6	The Simulation of Tools	164
7.6.1	The Explanation of Problem	165
7.6.2	AGG Simulation	166
7.6.3	The Rules and their Policies of the WSMS System	167
7.6.4	The Result of Execution Rules	173
7.7	Summary	174
8	The Extension of Web Services to Support PHP Moodle by Using NuSOAP Technique	175
8.1	Introduction	175
8.2	PHP Web Service Implementations	177
8.3	NuSOAP (SOAP Toolkit for PHP)	179
8.3.1	The Installation of NuSOAP Package	180
8.3.2	The Architecture of NuSOAP Package	181
8.4	Introducing the Web Services Approach to Meet VLE (Moodle) . . .	182
8.5	The Implementation of Web Services to Support Moodle by Using NuSOAP Approach	186
8.5.1	Using WSDL	188
8.5.2	Complex Types for WSDL	188
8.5.3	Registering Functions for WSDL	190
8.5.4	The WSDL File	191
8.5.5	Implementing Functions	192
8.5.6	Database Services	192
8.5.7	Login Services	194

8.5.8	Student Service	197
8.6	Summary	198
9	Analysis and Evaluation of the WSMS Model for VLEs	199
9.1	Introduction	199
9.2	The Value of the Research Study	200
9.2.1	The Research Community	201
9.2.2	Teachers	203
9.2.3	Learners	204
9.3	Analysis of Models for VLE	206
9.3.1	Course and Assignment Services	206
9.3.2	Main Page	207
9.3.3	Login Service	208
9.3.4	Admin Service	208
9.3.5	Teacher Service	210
9.3.6	Student Service	210
9.3.7	Guest Service	211
9.4	Evaluate the Models for VLE	211
9.4.1	Questionnaire	213
9.5	Summary	227
10	Conclusion & FutureWork	228
10.1	Future Work	234
10.1.1	Security and Privacy	236
10.1.2	Trust	237
10.1.3	Flexibility	237
	References	237
	Appendices	261

A Publications	262
A.1 Journals	262
A.2 Conferences	262
B Glossary of Acronyms Used	265
C The Architecture of NuSOAP Package	268
C.1 nusoap_base Class	268
C.2 soap_fault Class	269
C.3 nusoap_XMLschema Class	269
C.4 soapval Class	270
C.5 soap_transport_http Class	271
C.6 nusoap_server Class	271
C.7 WSDL Class	272
C.8 nusoap_parser Class	272
C.9 soapclient Class	273
D Rules	274
D.1 Rule 1	274
D.2 Rule 2	275
D.3 Rule 3	275
D.4 Rule 4	276
D.5 Rule 5	276
D.6 Rule 6	277
D.7 Rule 7	277
D.8 Rule 8	278

List of Tables

2.1	SOAP vs. XML-RPC	32
2.2	XML elements of WSDL	38
3.1	Functional and Non-Functional Requirements of Virtual Learning En- vironments	53
3.2	The Main Aspects for Evaluating Virtual Learning Environments . .	54
3.3	The Main Features and Tools of Virtual Learning Environments . . .	57
3.4	Indicative List of Open Source Management Systems	62
4.1	General Information about the Selected Products	69
4.2	Summaries of the Features and Capabilities of VLE tools	70
4.3	The Comparison between Selected VLE Products based on Learner Tools.	71
4.4	The Comparison between Selected VLE Products based on Support Tools.	72
4.5	The Comparison between Selected VLE Products based on Technical Specifications Tools.	73
4.6	The Final Result of the Comparison between Ten VLE Products. . .	74
4.7	Comparison based on focusing on the technical aspects of the VLE systems	76

4.8	Comparison based on focusing on some Features and Categories of VLEs.	77
4.9	Results of the Adaptation Category	77
4.10	The Users in Moodle Software	87
4.11	The Tools in Administration Form in Course in Moodle Sites	88
4.12	Advantages and Disadvantages of Forums in VLEs	90
4.13	The number of websites using Moodle in some countries around the world.	96
4.14	The Moodle Sites in Saudi Arabia in 2008.	98
6.1	Features of users in WSMS model	141
7.1	The comparison between some policy languages	152
7.2	The Syntax and Expression of Ponder Policy	156
7.3	The Domain Paths	163
8.1	The details of courses and assignments	185
8.2	The location of details of courses and assignments (table, field and type)	186
8.3	The Files in WSMS System and all Schools	187
9.1	The Final Result for all Questions of the Questionnaire for all Users .	225

List of Figures

1.1	The Outline of Thesis	14
2.1	The General Architectural Model of Web Services	22
2.2	The Scenario and Model of Web Services	24
2.3	Conceptual Architecture to adapt LISs as Web Services	25
2.4	The Web Services Technology Stack	26
2.5	The structure of a SOAP document	30
2.6	XML-RPC Structure	31
2.7	The Structure of Type Element in WSDL	34
2.8	The Structure of Message Element in WSDL	34
2.9	The Structure of Port Type Element in WSDL	35
2.10	The Structure of Binding Element in WSDL	36
2.11	The Structure of Ports Element in WSDL	37
2.12	The Structure of Service Element in WSDL	37
2.13	The Structure of Operation Element in WSDL	38
3.1	VLE as a Sub-System within the E-learning Environment	51
3.2	User-experience Architecture [1]	52
3.3	Structure of Open Source Platform [2]	59
3.4	The Future of VLEs - The Visual Version [3]	64
4.1	The Total Features of the Ten VLE Products	75

4.2	Chart of use of web-based e-learning systems in Universities	78
4.3	Architecture of Moodle Sites	85
4.4	The Main Screen	86
4.5	The Main Screen of Assignment Module in Moodle	92
4.6	Assignment Types in Moodle	93
4.7	Online Text Assignment Type in Moodle	93
4.8	Upload Single File Assignment Type in Moodle	94
4.9	Advanced Uploading of File Assignment Type in Moodle	94
4.10	The Architecture of Assignment Module in Moodle	95
4.11	Moodle Sites that have registered with Moodle across the world. . . .	97
4.12	the increasing use of Moodle in Saudi Arabia over six years	97
5.1	The Proposed Architectural Extension Based on Web Services for VLE	106
5.2	UML for representing Web service offers and requests	107
5.3	Web Services in Multi Moodle Platforms	109
5.4	Complex Data Type for Course	110
5.5	Complex Data Type for Assignment	111
5.6	Register getCourses Services in Server	111
5.7	Courses connects with database (SQL)	112
5.8	The details of the getCourses service after registering with NuSOAP server	113
5.9	The Levels of Authorisation in WSMS	117
6.1	The Architecture for an SOC-based VLE (Moodle)	127
6.2	Client/Server in VLE (Moodle)	129
6.3	Presentation logic in VLE (Moodle)	130
6.4	Business Logic in VLE (Moodle)	131
6.5	Separating Presentation logic and Business logic	132
6.6	VLE after Abstraction	132

6.7	Service Oriented Style, Syntax of UML for Web services, and semantics of the Find operation, Service Description and Requester	133
6.8	Communication Diagram of Web services Based on VLE	135
6.9	Ontology for Registration of Module	136
6.10	Provider Rules	137
6.11	Requestor Matching the Specifications	138
6.12	The scenario of Web Services for WSMS with two VLE Softwares (Moodle)	139
6.13	Four Users in Proposed Model (administrator, teacher, student and guest)	140
6.14	Basic model of Finite State Machine for Web Services	144
6.15	The States of WSMS by using Finite State Machine	146
6.16	WSMS using Step by State in Finite State Machine	147
6.17	WSMS using Multiple Run Test in Finite State Machine	148
6.18	Trackback of Accept Test for WSMS Using Multiple Run Test	148
7.1	Authorisation Policy Syntax	158
7.2	Filters on Positive Authorisation Actions	159
7.3	Delegation Policy Syntax	159
7.4	Refrain Policy Syntax	160
7.5	Obligation Policy Syntax	161
7.6	The Search paths in Domains	162
7.7	Set Operation	163
7.8	The Syntax of WSMS Policy	165
7.9	The Relationship between users and course in VLE (Moodle)	165
7.10	The Type Graph of WSMS	166
7.11	Rule 1-The enrolment of a student	168
7.12	The Policy of Rule 1	168

7.13 Rule 2 - the login attempts of a student	169
7.14 The Policy of Rule 2	169
7.15 Rule 3 - the binding to access the assignment service	170
7.16 The Policy of Rule 3	171
7.17 Rule 4 - the binding to access the course service	171
7.18 The Policy of Rule 4	172
7.19 Rule 5- Teacher manages course	172
7.20 The Policy of Rule 5	173
7.21 The Result of Execution Type Graph of WSMS System	173
8.1 The Architecture of NuSOAP Package	181
8.2 The WSDL configuration for a NuSOAP server	188
8.3 The Code ComplexType for Course and ArrayOfCourses	189
8.4 The Code ComplexType for Assignment and ArrayOfAssignments . .	190
8.5 The Registration of getCourses Method	190
8.6 The Registration of getAssignment Method	191
8.7 The functions that have registered as Web services in WSMS system .	191
8.8 Courses connects with database (SQL)	192
8.9 Data Source and How It Connects with Database	193
8.10 The Select Query of searchCourse Service in WSMS	193
8.11 The Relationship of the Select Query of searchCourse Service in WSMS	194
8.12 The Context Definitions in Moodle	194
8.13 Login Form in WSMS	195
8.14 The Complex Type for login services	195
8.15 The Registration for login services in NuSOAP	196
8.16 The Session to Store User's Data	196
8.17 The Registration of Student Service in Server (NuSOAP)	197
8.18 The Complex Type of Student Service	197

9.1	The Main Page of WSMS where all users can search and see basic information	207
9.2	The Front Page of WSMS	208
9.3	The Main Page of Admin, Teacher and Student Users	209
9.4	The Student Details that Admin can see	209
9.5	The Student Details that Student can see	210
9.6	The Message that appear for users if they do not have access permission	210
9.7	The results of answers of Question 1 by guests	213
9.8	The results of the answers to Question 2-a by guests	214
9.9	The results of the answers to Question 2-b by guests	214
9.10	The results of the answers to Question 2-c by guests	215
9.11	The results of the answers to Question 1 by students	216
9.12	The results of the answers to Question 2-a by students	216
9.13	The results of the answers to Question 2-b by students	217
9.14	The results of the answers to Question 2-c by students	217
9.15	The results of the answers to Question 3-a by students	218
9.16	The results of the answers to Question 3-b by students	218
9.17	The results of the answers to Question 1 by teachers	219
9.18	The results of the answers to Question 2-a by teachers	220
9.19	The results of the answers to Question 2-b by teachers	220
9.20	The results of the answers to Question 2-c by teachers	221
9.21	The results of the answers to Question 3-a by teachers	221
9.22	The results of the answers to Question 3-b by teachers	222
9.23	The results of the answers to Question 1 by administrators	223
9.24	The results of the answers to Question 2-a by administrators	223
9.25	The results of the answers to Question 2-b by administrators	224
9.26	The results of the answers to Question 2-c by administrators	224
9.27	The Final Result of the Questionnaire for all Users	226

10.1 The Security Environment of a Collaborative E-learning Model using a Web services technology	235
C.1 The Methods in nusoap_base class	269
C.2 The Methods in soap_fault class	269
C.3 The Methods in nusoap_xmlschema class	270
C.4 The Methods in soapval class	270
C.5 The Methods in soap_transport_http class	271
C.6 The Methods in nusoap_server class	272
C.7 The Methods in wsdl class	272
C.8 The Methods in nusoap_parser Cclass	273
C.9 The Methods in soapclient Class	273
D.1 Rule 1-The enrolment of a student	274
D.2 Rule 2 - the login attempts of a student	275
D.3 Rule 3 - the binding to access the assignment service	275
D.4 Rule 4 - the binding to access the course service	276
D.5 Rule 5- Teacher manages course	277
D.6 Rule 6- Teacher manages course	277
D.7 Rule 7- Binding to access assignment	278
D.8 Rule 8- managing course and supervisors teacher	278

Chapter 1

Introduction

Objectives

- To describe the motivation, aims and objectives behind SOC and VLEs.
 - To identify the research questions and highlight on original contributions.
 - To explain thesis structure and reading guide.
-

1.1 Background

Technological development has become so manifest in all fields that it is now crucial to take it into consideration whenever possible. Such development has resulted in the tremendous amount of information that exists nowadays. Hence, it is not unusual to meet the expression ‘the information revolution’ wherever you go.

Service Oriented Computing (SOC) has become an active area of research and development. Web services, as an example of Service-Oriented Architecture (SOA), which is a part of SOC, support the integration of software applications in an incremental way, using existing platforms and languages that utilize and adopt existing

legacy systems. Virtual Learning Environments (VLEs) are one of the emerging domains that need to be extended to SOA.

Using the Internet to support teaching and learning has become a trend in modern higher education by using VLEs that support online learning and teaching over the Internet. These VLEs feature a set of robust core capabilities that enable teachers to efficiently manage courses and assignments, and foster collaboration, among other key functions. The next generation of VLEs could address the limitations of current online systems by providing a richer social context for online learning.

The main aim of this research is to extend the core idea behind the VLE tools that are gradually dominating academic institutes. It is important to know that educational theory is outside the scope of the thesis. However, aspects of educational theory guided the architectural design and develop of our VLEs.

1.2 Motivation of the Research

It is undoubtedly the case that the emergence and use of computers can be regarded as the main scientific and technical achievement of the present century, and it has given a great boost to scientific and industrial progress. However, VLEs as Web services are still not generally accessible for academic institutions. Developers are still developing VLE services and they continue to be adopted by some scientific projects. Nowadays, there is an increasing demand for VLE methodologies and technologies. This part will summarize motivations in three main points, which are:

1. VLEs as Web services are still not generally accessible for academic institutions. Therefore, VLEs need:

- To demonstrate that VLEs can be considered as services that can be published, discovered and composed as perceived in the SOC paradigm.

This project should open a wide door for collaborative VLE services that

are widely distributed, more flexible and effective.

- To modify the traditional security requirements to cater for the highly mobile and changeable environment of VLEs.
2. To find the most suitable technology for supporting the worldwide trend of offering online joint courses over the Internet, which includes institutes in different countries who employ roaming staff and target mobile students. This will allow teachers at different institutions in different countries to work together and share material that is hosted on different VLE platforms.
 3. There are limitations for using VLEs in Saudi Universities, especially in Qassim University; despite the large number of VLE platforms that exist in institutions in Saudi Arabia, they are not used in a well-organised or well-integrated way. Most of these VLE platforms are only used as local self-contained systems. The use of VLEs through the Internet in Saudi Arabia is not fully developed, and this has reduced the benefits of sharing and exchanging data among different institutes in Saudi Arabia. Therefore, this thesis will try to find the most suitable VLE software that meets Qassim University's needs.

1.3 Aim & Objectives of the Research

The main aim of this study is that using Web services in VLEs should allow educators at different institutions in different countries to work together and share material by connecting individual courses together, even when they are hosted on different VLEs. Therefore, educators at different institutions who teach the same course will be able to share courses, assignments, chat, etc., and be able to communicate with each other. The main objectives of this investigation are:

1. This thesis aims to explore novel architectural designs for VLEs based on the SOC paradigm by extending the core idea behind VLE tools that are becoming

so widespread in academic institutes. It aims to build a VLE around the Web services technology. This thesis aims to present new technologies for VLEs that no previous works have done, and these techniques are:

- Introducing a Web services technology to meet the requirements of VLE by extending the existing architecture of VLE. This study aims to allow institutions to exchange data cheaply and easily, and to deploy techniques such as service descriptions, registrations and discovery for collaborative VLE services, all of which are flexible, easily accessed and effective.
 - This study aims to build a new extension to the structure of Web services: the Web Services Matching and Selection (WSMS) system. It involves discovering a set of semantically equivalent services by filtering a number of available services and then selecting a suitable service during execution.
2. This study considers that VLEs can be services, which can be created, published, discovered and composed within the SOC paradigm. This allows teachers at different institutions in different countries to work together and share material by connecting individual courses together, although hosted on different VLE platforms. Therefore, any number of teachers who deliver the same course at different institutions can share course materials, assignments, exams, etc., and can readily communicate with each other.
 3. To devise a policy-based technique to enforce security requirements for VLEs. This research aims to modify the traditional security requirements to cater for the highly mobile and changeable environment of VLEs. This objective will be achieved through policy-based techniques.
 4. To build a test-bed for a VLE that is based on the Modular Object-Oriented Dynamic Learning Environment (Moodle) platform.

1.4 Research Questions

The overall and central questions investigated by this thesis are:

1) How can we enable Web services to facilitate the process of building VLE around SOC?

2) In what ways can we enhance the traditional security requirements and modify them to cater for the highly mobile and changeable environment of VLE?

In order to be able to answer these questions, we define a set of research questions that addresses the problems in detail:

1. What is the best technology for supporting the worldwide trend of offering online joint courses over the Internet, which includes institutes in different countries who employ roaming staff and target mobile students?
2. Can this technology be applied to all VLEs, and especially to Moodle?
3. And what are the effects by applying Web services to VLEs?
4. In what ways can we demonstrate that VLEs may be considered as services that can be fully utilised and published within the SOC paradigm?

1.5 Research Methodology and Evaluation

This thesis focuses on the integration of two important fields in computing, namely VLE and SOC. VLEs are more and more becoming a major part of the strategy for delivering online and flexible learning. Hence, there is a growing demand for methodologies and technologies that support them. The most important challenge

of this investigation is to find a suitable technology to develop and to aid the distribution of VLE software packages and related technology. This section focuses on the research methodology and evaluation as follows:

1.5.1 Methodology

In this research, we demonstrate that VLEs can be considered as services that can be published, discovered and composed as perceived in the SOC paradigm. We have succeeded in using Web services as a technique to apply SOC with the VLE domain, which are using Moodle as an example of VLEs. Figure 6.12 in Chapter 6 has more details on how Web services work within the VLE domain. The methodology to achieve this project consists of two main parts. This section will summarize these parts and outline why this thesis uses them; these parts are:

1. The first part of our methodology is that this thesis has built a new extension to the current architecture of Web services as in Figure 5.1 in Chapter 5. This extension has extended to meet the technical requirements of VLE environment, as in Chapters 5 and 8. The idea behind this extension is to provide a method of applying Web services, which is a technique for applying SOC to VLE as an application domain. One of the greatest advantages of Web services is that they allow a number of applications to be integrated more quickly, easily and cheaply than ever before. They are expressed as WSDL, which is an XML-based language. A service specifies a contract between the client's requests and the operations that it can offer. A service may be published and discovered using UDDI, while SOAP allows vendor-neutral communication between applications over HTTP. The main reasons for choosing the Web services technology are:

- Web services allow pieces of software to be written in different languages and to run on different operating systems, both cheaply and easily;

- They allow applications running in different parts of an organization and in different organizations to exchange data easily and cheaply;
 - They are applications running anywhere on any technology that have Web services.
2. The second part of our methodology is that this study has built a new extension to the current architecture of VLEs as in Figure 4.3 in Chapter 4. This extension has a new layer that provides a generic solution for migrating any stand-alone into SOC/VLE. It allows the owners of services to publish a description of their services and controls access to them. In addition, it works as a central store that makes possible service discovery by service requestors. It provides a searchable repository of service descriptions where service providers publish their services, and where service requestors find services and obtain binding data for these services. This thesis uses VLEs as an application domain, and Moodle is an example of this domain. Moodle is a VLE that lets teachers provide and share documents, graded assignments, etc. with their students in an easy-to-learn manner, and to create quality online courses. The main reasons for choosing Moodle over other VLE products in this study are.
- Moodle is free, which means users are free to download, use, modify and even distribute it under the terms of GNU.
 - Moodle can be used on almost all servers that can use PHP, and data is stored in a single database: MySQL, Oracle and others.
 - The credibility in Moodle is high - there are more than 50,000 sites from 193 countries that have registered. It has a strong grounding in social constructionist pedagogy.
 - It works well with 75 languages in 193 countries. It runs without modification on Unix, Windows and any system that support PHP.

1.5.2 Evaluation and Testing

The evaluation in this thesis was conducted along two paths:

a) Building a Test-bed

This thesis is built a test and deploys it at two schools. These schools are hosted in different places, and they are:

1. Saudi School in Leicester

This thesis has used Saudi School in Leicester (SSiL) as test-bed environment. SSiL is hosted in (www.saudi-school-le.net/), which uses Moodle as a VLE platform. We consider this school as Service Provider 1 (SP1) in our model (WSMS), as in Figure 6.12 in Chapter 6. This provider publishes its services in service registry. When the service requestor searches for his/her specific order, the WSMS will reduce the service selection by using the attributes of both consumer and service, and offer the suitable one that meet requestor's needs.

2. Ajlan's High School

We have used Ajlan's High School (AHS) as test-bed environment with SSiL, but AHS is hosted in different places (www.ajlan-alajlan.com/moodle/). We consider this school as Service Provider 2 (SP2) in our model (WSMS), as in Figure 6.12 in Chapter 6. Also, this provider will publish its services in service registry to enable service requestor to search in service registry to find the suitable service that meet his/her needs.

b) Questionnaire

This study has also used the questionnaire method to analyse and evaluate the proposed solution in this thesis, which is the WSMS system. The overriding aspect of the analysis and evaluation of the WSMS system is to gain a greater understanding

of the needs of the users in order that the WSMS system is better able to meet those needs in future.

The questionnaire is based on a sample of administrators, teachers, students and guests. The answers to the questions are a choice of: strongly agree, agree, normal, disagree, and strongly disagree. The questions depend on the kind of user and we have tried to set these questions to consider that kind of user.

The questionnaire was sent to some users who were available at SSiL in United Kingdom. This school has 254 students, 21 teachers and 6 administrators. The users in this school are from various countries such as Saudi Arabia, Libya, UK, Somalia, Syria, Palestine, Egypt, Algeria, Morocco and the Gulf countries. In addition, the questionnaire was sent to some parents who were available in Leicester city in United Kingdom. In this method, the researcher worked with about 200 samples (130 students, 46 guests, 18 teachers and 6 administrators).

1.6 Contributions of the Thesis

This investigation has found that the most suitable technology for developing and distributing VLEs is by using Web services technology and related techniques. Web services together with VLE enable users on different VLE systems in different countries to work together and share material. As this study has chosen the Web services technology, it has also chosen Moodle as an example of VLE platforms.

The study presents many contributions to the knowledge by developing VLEs through Web services technology. Moreover, this thesis presents new technologies for VLEs that no previous works have done before, and these technologies are:

1. The first technology that this thesis presents is that this study has built a new extension to the existing structure of Web services: the Web Services Matching and Selection (WSMS) system, as in Figure 5.1 in Chapter 5. This

system involves discovering a set of semantically equivalent services by filtering a number of available services and then selecting a suitable service during an execution. The proposed approach has extended the current architecture of Web services to meet the technical requirements of Moodle, as an example of VLEs, as in Chapters 5 and 8. This approach allows institutions to exchange data cheaply and easily, and deploys techniques such as service descriptions, registrations, discovery and binding for collaborative VLE services that are distributed in a flexible and effective manner, as in Figure 6.12 in Chapter 6.

2. The second technology that this thesis presents is that this study has built a new extension to the existing architecture of VLEs: Web Services Technology, as in Figure 4.3 in Chapter 4. This new layer provides a generic solution for migrating any stand-alone into SOC/VLE. This technology allows the owners of services to publish a description of their services and controls access to them. In addition, it works as a central store that makes possible service discovery by service requestors. It provides a searchable repository of service descriptions where service providers publish their services, and where service requestors find services and obtain binding data for these services, as in Chapter 8.
3. The traditional security requirements of VLEs have been modified to cater for the highly mobile and changeable environment of VLEs. This employs a policy-based technique (Ponder policy), and we build a test-bed to prove the security requirements for VLE based on SOC, as in Chapter 7.

In addition to the above contributions, there are some other important contributions to the knowledge that this thesis presents. These contributions depend on the above contributions and they are:

1. This project has allowed educators at different institutes in different countries to work together and share material by connecting individual courses together,

even though they are hosted on different VLEs. This reduces the time and effort needed by both teachers and students. In addition, this enables better coordination and cooperation between academic institutions than ever before.

2. Another contribution to the knowledge is that this research has laid a good foundation for other researchers intending to work in this area by providing a full picture of Web services, VLEs and their components and related issues.
3. This study is aimed at taking the right decision when choosing a suitable VLE platform to meet the requirements of Qassim University, as in Chapter 4. This is a large university, and so it needs a strong VLE that meets all its needs and this research will hopefully aid other universities as well as Qassim University in the search for the best VLE platform.
4. These contributions are published in learned journals and conferences.

1.7 Thesis Structure and Reading Guide

This thesis contains a survey of the VLE and SOC technologies, a description of the work, and an overview of the results and contributions. An outline of the structure of this thesis is:

Chapter 2: Service-Oriented Computing

The second chapter provides an overview of the literature on SOC and its benefits. In addition, it focuses on Web services, including the benefits, architecture and scenario of Web services. Also, it describes the Web Services Protocol Stack in detail, including the XML, WSDL, SOAP and UDDI languages, and shows some examples of how XML is used in the real world. What Web services are good for, and the future of Web services are also mentioned briefly at the end of this chapter.

Chapter 3: Virtual Learning Environments as E-learning Systems

This chapter presents an outline of the literature on VLEs and their benefits. The architecture, challenges and functional and non-functional requirements of VLEs are also described on this chapter. The main features and tools of VLEs are described in more detail in this chapter, together with the benefits and limitations of OSSs and why they are used. The future of VLEs is mentioned at the end of this chapter.

Chapter 4: ‘Why Moodle?’

This chapter is an important chapter because it focuses on the system that this thesis has used to apply its approach. This system is Moodle and this chapter gives an overview of the literature of Moodle including the reasons for choosing Moodle and its architecture. The most important section in this chapter is the comparative study between Moodle and other VLE products. Also, this chapter presents the tools and activities of Moodle and concentrates on the assignments module as a specific area of study. The number of websites using Moodle all over the world is also mentioned in this chapter.

Chapter 5: Production of Technical Requirements for SOC-Based on VLE

In this chapter, we introduce a production of technical requirements for an SOC-based VLE by highlighting the requirements (functional and non-functional) for VLEs (Moodle) based on Web services. At the beginning of this chapter we give a brief summary on the distribution systems of VLEs. The main part of this chapter contains an example: Web services for Moodle, and then the scenario of WSMS and its requirements to enable Moodle to use Web services.

Chapter 6: The Architecture for an SOC-Based VLE

The sixth chapter contains the general architecture for an SOC-based VLE. This architecture will move Moodle from its current Web applications to new Web applications and will open the door for other VLE products to use Web services as a new Web application. The UML language is described at the beginning of this chapter as our architecture is built by using this language. The architecture of WSMS with two VLE platforms is described in Section 5. The scenario of this architecture will be described briefly by using the FSM tool at the end of this chapter.

Chapter 7: Devise a Policy-Based Technique for Enforcing the Security Environment of VLEs

Ponder policy is described in detail in this chapter and it has two main parts. The first part is ‘access control’ policy and the second part is ‘obligation’ policies. Policy specification languages are described in Section 4. At the end of this chapter, we explain the problem of tool simulations and of AGG simulation by set some rules.

Chapter 8: The Extension of Web Services to Support PHP Moodle by Using the NuSOAP Technique

The PHP Web services implementation methods are described at the beginning of this chapter. The next section focuses on NuSOAP as the most well-known technique. The main parts of this chapter introduce the Web services approach to meet VLE (Moodle), and the implementation of Web services to support PHP Moodle by using the NuSOAP approach.

Chapter 9: Analysis and Evaluation of the WSMS Model for VLEs

This chapter analyses and evaluates the final models of VLE with a Web services technology. At the beginning of this chapter, we introduce the value of the study

for the research community, teachers and learners. We then analyse the final VLE models with a Web services technology. In the last part of this chapter we evaluate the final VLE models with a Web services using the questionnaire method.

Chapter 10: Conclusion and Future Work

The tenth chapter summaries the work presented in this thesis. The significance of the main findings is presented. In addition, it highlights the most important contributions, and then discusses methods and directions for possible future studies.

The reading guide of this investigation is illustrated in Figure 1.1 below:

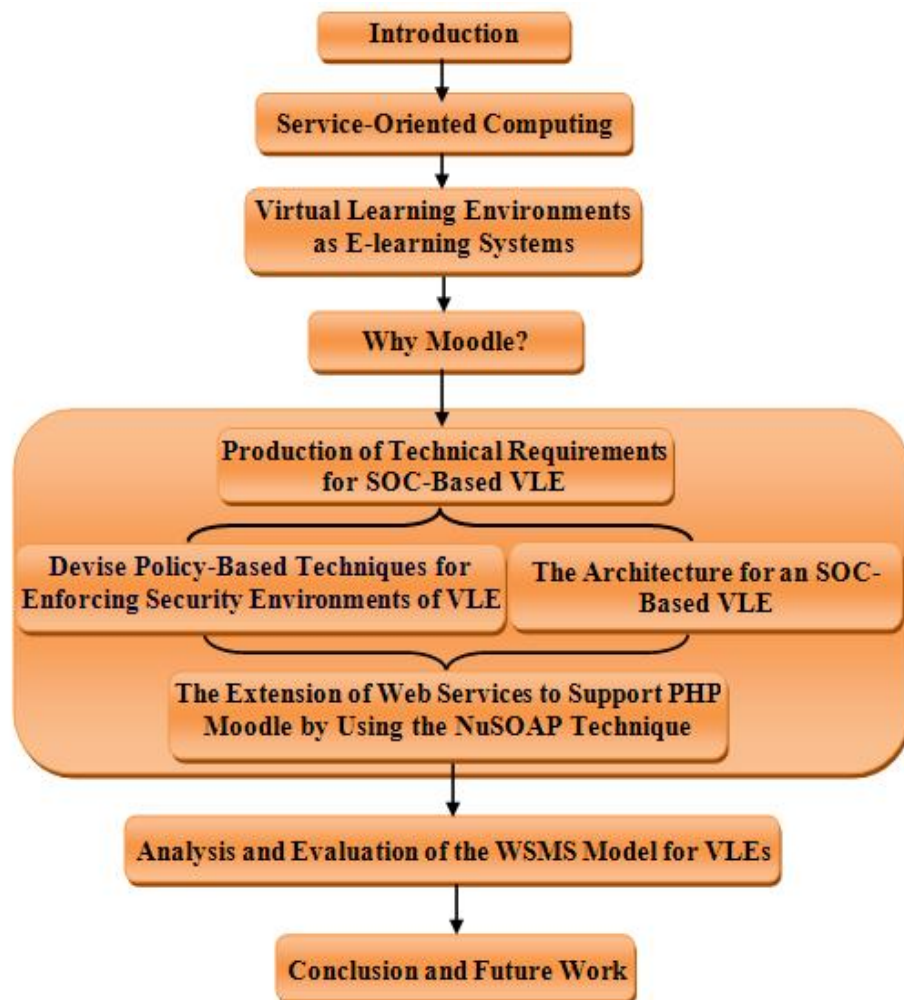


Figure 1.1: The Outline of Thesis

Chapter 2

Service-Oriented Computing

Objectives

- To provide an overview of literatures on SOC and its benefits.
- To focuses on Web services (benefits, architecture and scenario).
- To describe Web Services Protocol Stack.
- To give an overview on What are Web services good for, who is using Web services and the future of Web services in brief.

2.1 Introduction

Service-Oriented Computing (SOC) has become an active area of research and development, as e-businesses are continually evolving toward SOC-type architecture. SOC has many advantages that help IT and managers to develop and manage their systems [4, 5]. One of the most important advantages of SOC is their use in machine-to-machine interaction [6]; they allow applications to be integrated faster, more easily and more cheaply than ever before. This integration occurs at an advanced stage

in the protocol stack, based on messages centred more on service semantics and less on network protocol semantics, which enables the loose integration of business functions [4, 7].

Nowadays, the ability to exchange data between internal business units, partners, and customers is essential for success. A number of continuing research efforts focus on a variety of aspects of SOC, particular Web service, including composition, specification modelling, discovery and verification [8]. Web services are the most promising technology for easier system integration by providing standard protocols through using XML messages for data exchange and a standard interface declaration language [9, 10, 11]. Over the last couple of years, Web services have expanded to become more popular with application developers. They can be developed on any computer platform and in any development environment, as long as it can communicate with other Web services using these common protocols [12, 13, 14, 15].

This chapter presents an outline of the literature on SOC and its benefits. In addition, it focuses on Web services and related techniques, including the benefits, architecture and scenario of Web services. Also, it describes the Web Services Protocol Stack in detail, including the XML, WSDL, SOAP and UDDI languages, and shows some examples of how XML is used in the real world. What Web services are good for, who is using Web services, and the future of Web services are also mentioned briefly at the end of this chapter.

2.2 Service-Oriented Computing

The promise of SOC is that it allows services to be loosely coupled, creating flexible dynamic business processes and agile applications that can span organizations and computing platforms [16]. Thus SOC exploits the potential of business and educational services, making them essential elements in developing applications and solutions. However, basic SOC's do not address overarching concerns such as man-

agement, service orchestration, service transaction management and coordination, security, and other concerns that apply to all components in service architecture. Figure 2.1 illustrates the roles and sequence of events in an SOC [5, 17, 18].

2.2.1 Definition of Service-Oriented Computing

SOC is a newly emerging paradigm for distributed computing and e-business processing that utilizes services as essential elements to allow the building of agile networks of collaborative business applications distributed within and across organizational boundaries [19, 20]. It is a computing paradigm that uses services as essential elements for developing applications to build service models. It depends on SOA, which is a way of reorganizing software applications and infrastructure into a set of interacting services. The definition of SOC is given by [4, 18, 21] as follows:

‘SOC is a process of discovering and composing the suitable services to satisfy a specification. It is a computing paradigm for distributed computing that is changing the way software applications are designed, delivered, consumed and architected. It involves extended, loosely coupled activities among two or more autonomous business partners’[4, pp.43].

SOC refers to the set of methods, principles, and concepts that represent computing in SOA in which software applications are constructed based on independent component services with standard interfaces [22]. The most important idea of SOC is to separate software engineering from programming, to highlight software engineering, and to de-emphasize programming. SOC separates software development into three parties: service providers by programmers, application builders by software engineers, and service brokers [23, 24].

All major computer companies, government agencies, banks, airlines and travel agencies have moved towards SOC and have adopted it. In addition, universities

around the world have geared their computing research toward SOC-based modelling languages, service verification, validation and automated code generation [23].

2.2.2 Benefits of Service-Oriented Computing

There are a number of research challenges of SOC that need to be addressed including, among other things, integration, monitoring, composition, discovery of services and their quality, development, evolution and security. These are attracting the interest of researchers in different communities, including those involved in databases, software engineering, artificial intelligence and distributed systems. SOC provides the following benefits [4, 9, 20, 23, 25]:

1. Allows separate software engineering from programming into service providers by programmers, application builders by software engineers, and service brokers, in order to focus software engineering, and to de-emphasize programming;
2. Enables novel flexible business applications of open source systems that would not be possible otherwise;
3. Enables the customization of new applications by providing a Web service interface that eliminates messaging problems, and by providing a semantic basis to customize the functioning of the application;
4. Improves the productivity of programming and administering applications in open source systems; such applications are notoriously complex;
5. Enables the efficient usage of grid resources and facilitates utility computing, especially where redundant services can be used to achieve fault tolerance;
6. Provides a semantically rich and flexible computational model, for which it is easier to produce software.

2.3 Web Services

Web services are rapidly emerging as a popular standard for sharing data and functionality between loosely-coupled and heterogeneous systems. Currently, most organizations employ disparate applications that exchange and store data in different approaches [26]. They support the basics of e-business processes that are distributed over the Internet and that exist via standard protocols and interfaces [8]. They are an implementation of SOC that provide a simple mechanism to connect applications regardless of device or technology. They have evolved as a practical, cost-effective solution for uniting information distributed between critical applications over language barriers, operating systems and platforms that were impassable [21].

Web services are a group of emerging and established communication protocols that consist of Extensible Markup Language (XML), Simple Object Application Protocol (SOAP), Universal Description Discovery and Integration (UDDI) and Web Services Description Language (WSDL) over Hypertext Transfer Protocol (HTTP) [12, 27]. They allow applications to be integrated faster, more easily and more cheaply than ever before. They are expressed as a WSDL that is XML-based language [21]. Services specify a contract between the client and the operations that can be expected. Services maybe published and discovered using UDDI, while SOAP allows vendor-neutral communication between applications over HTTP [14, 15, 28, 29].

2.3.1 Definition of Web Services

Web services, also called service computing, are uncomplicated and self-contained applications which achieve functions from easy requests to difficult business processes. They happen through a machine-to-machine communication without a user interface to call the services. The definition of Web services is given by [5, 13] as follows:

‘Web services are a new breed of Web application. They are self-contained, self-describing and modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. ... Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service’ [5, pp.1 and 13, pp.21].

The WWW Consortium [30] has defined Web services as a software application identified by Uniform Resource Identifier (URI), whose interfaces and binding are capable of being defined, described and discovered by XML artefacts, and which supports direct interactions with other software applications using XML-based messages via Internet-based protocols.

Web services are software components that communicate using pervasive, standards-based Web technologies including HTTP and XML-based messaging [19]. They represent the efforts of researchers and developers to extend the Web from an infrastructure that provides services to humans to one that provides services to software looking to connect with other software. They can be developed on any platform and in any development environment, and can communicate with other Web services using common protocols [5, 31, 32].

2.3.2 The Benefits of Web Services

Web services are based on industry standard protocols with universal vendor support, which can exploit the Internet for low-cost communications, for high-speed implementation processes, and for improved transport mechanisms [33]. The ambiguous messaging approach provides connections and information sharing scenarios via services that are self-describing and can be discovered automatically. Using Web services can provide the following benefits [12, 15, 20, 34, 35, 36]:

1. Web services allow pieces of software to be written in different languages and to run on different operating systems, both cheaply and easily;
2. They allow applications running in different parts of an organization and in different organizations to exchange data easily and cheaply;
3. Collaborations of Web services depend on runtime to enable the just-in-time (JIT) technique for improving the runtime performance of computer program integration successfully. Service consumer describes the service required and uses the service register to discover and select the optimal service;
4. They provide a much richer specification of the service compared to previous technologies, which can be accessed programmatically; the time taken for developers to properly understand how to use an existing interface slows down the time that new connections can be established;
5. Web services promise to greatly increase interoperability and ease data exchange even as it lowers costs. That is an ambitious agenda, but if Web services achieve even moderate success, the impact on the business world will be profound.
 - Web services offer a way to maintain and integrate legacy systems at a lower cost than typical Enterprise Application Integration (EAI) efforts.
 - Web services reduce the costs and complications of having multiple platforms, running on everything from mainframes to servers, and from desktops to Personal Digital Assistant (PDAs).
6. Web services are applications running anywhere on any technology or device that have Web services.
7. They utilise automated discovery and provide a mechanism for discovering service providers, which can also be automated.

8. Web services do not require using browsers or HTML.

Furthermore, several additional advantages can be provided by using Web services architecture: promoting interoperability by reducing the requirements for shared understanding, enabling the just-in-time integration technique, minimizing complexity by encapsulation, and enabling interoperability of legacy applications. Furthermore, Web services are built on simple standards that are used to publish applications such as SOAP (used to invoke Web services), WSDL descriptor (definition for Web services), and UDDI (the registry where Web services are located) [37, 38].

2.3.3 Web Services Architecture

The Web services architecture aims to provide a standards-based platform for SOC. It defines a set of specifications that support an open XML-based platform for the description, discovery, and interoperability of distributed, heterogeneous applications such as services [39]. Web services architecture is based on the interactions between three roles, as in Figure 2.1 [4, 24]: service provider, service registry and service requestor. The interactions involve publish, find and bind operations. Together, these roles and operations act upon Web services artefacts, the Web service software module and its description.



Figure 2.1: The General Architectural Model of Web Services

2.3.3.1. Service Provider

It is the owner of the service and it is responsible for publishing a description of its service to a service registry. It also hosts the service and controls access to it.

2.3.3.2. Service Registry (Broker)

It is a central store that makes possible service discovery by service requestors. This component provides a searchable repository of service descriptions where service providers publish their services, and where service requestors find services and obtain binding data for these services.

2.3.3.3. Service Requestor

It is a software component in search of the service to invoke across the Web. It finds the suitable service by discovering a set of available services that meets some pre-defined criteria [40].

- Publish: means that how the provider of Web services registers itself;
- Find: means that how an application finds a suitable Web service;
- Bind: means how an application connects to, and interacts with, a suitable Web service after it has been found.

2.3.3.4. The Scenario of Web services

The scenario of Web services consists of four stages and these stages are:

- The service provider deploys and publishes a description of its services to the service registry, and hosts these services and controls access to them.
- The service requester works with a service broker to discover the optimal services that meet and satisfy the specifications of the register.

- The service broker sends the services that have been found to the service requestor.
- In the final stage, the service requester negotiates with the service providers to bind the services after they have been found.

A service requester could be a human client, a device, an application, or any other Web service. A service broker provides registries that make Web services available. The service provider and requestor roles are logical constructs and a service can exhibit characteristics of both. Figure 2.2 illustrates these stages [41, 42, 43];

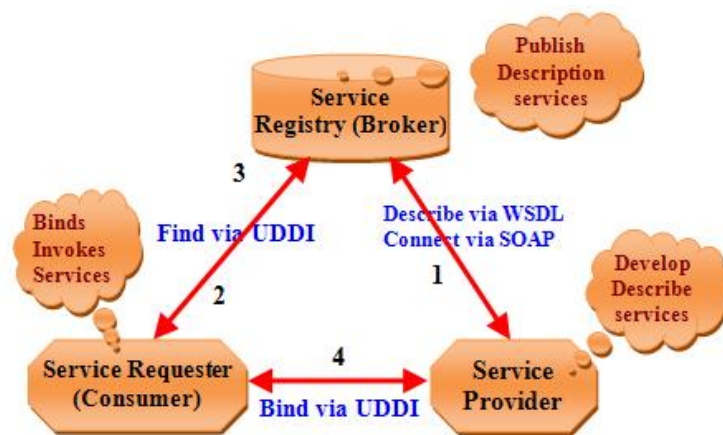


Figure 2.2: The Scenario and Model of Web Services

2.4 Legacy Systems

In the 1990s, Legacy Systems (LS) were written in COmmon Business-Oriented Language (COBOL). It is an application that used to be delivered to a company by an External Service Provider (ESP) [44]. A number of organizations have application programs that are currently working but they need to integrate them with other systems. Such computer systems are generally incompatible with other systems, are large and complicated to modify, and are costly to replace and develop [44, 45].

Brodie [46] defined a Legacy Information System (LIS) as any information system that significantly resists modification and evolution. These LISs have become significantly problematic to many organizations. The LISs run on old hardware that is slow and expensive to develop.

Currently, Web services represent a good solution for many LIS problems, which makes LISs reusable. Thus, a number of companies have started to adapt their legacy applications through Web services. Furthermore, the concept of Web services architecture together with the SOC approach makes LISs available for the Web, as in Figure 2.3 [46, 45].



Figure 2.3: Conceptual Architecture to adapt LISs as Web Services

2.5 The Web Services Technology Stack

Web services are a set of protocols based on XML/SOAP over HTTP [27]. Web Services Protocol Stack (WSPS) is a compilation of computer networking protocols that are used to implement, define and locate Web services interaction with each other. It is comprised of four main areas, as can be seen in Figure 2.4 [47, 41, 48]:

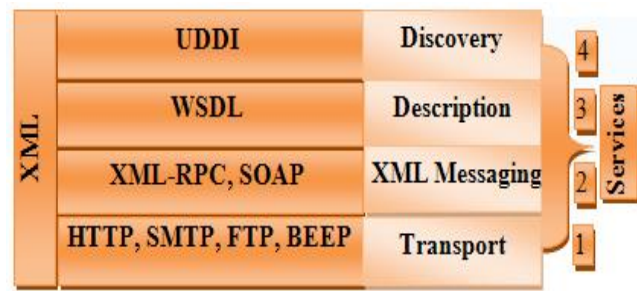


Figure 2.4: The Web Services Technology Stack

Many researchers and developers will be familiar with the WSPS, which has formed the initial specification for Web services. WSPS contains four services and XML as shown on the right of Figure 2.4. These services describe the best way to publish, find and bind services between the components of Web services. This section will give a brief explanation about these services, and we will start this section with the XML format as follows:

2.5.1 Extensible Markup Language

XML is at the heart of SOC, particularly Web services. It is used to exchange messages between services and applications by using SOAP (to describe the structure of messages by using XML Schema), and to describe Web services interfaces by using WSDL [49]. XML is centralized as recommended by the W3C standard. XML has been defined as a flexible way to create common information formats and share these formats and data across the Web. It is not only good for transmission of data from server to client, but it is also ideal for passing data from application to application

and machine to machine. XML standards have been implemented in every major operating system and programming language, ensuring that one of most important assets - data - will always be accessible in the future [50].

XML processing can invite significant runtime overhead in XML-based infrastructural middleware such as Web service application servers [51]. XML technologies allow user to future-proof Web sites and services. Today they can easily be adapted for future needs, allowing easy transformation into new data formats as they emerge, and building new services in order to accomplish new tasks with existing data. Additionally, XML serves as a common platform for the passing and sharing of data between different systems, allowing the rapid development of Web services that query, retrieve and share data between many sources [5, 12].

Advantages of XML

The good and bad of XML both lie in its verbosity such as well-formed pairs of named marking-up tags for XML elements, which contribute to its human friendliness and vendor-neutrality but which require extra computation in processing. The computation specific to XML includes variable representations for the same tag, handling of namespaces, tolerance for multi-character encodings, etc. Nevertheless, XML has many advantages, which make it a universal and famous language, and these advantages are as follows [14, 50, 51, 52]:

1. XML is programming language, extensible, independent and improve data typing and structure.
2. It is modular. By allowing documents to reference other documents, XML provides for modular designs and promotes reuse.
3. XML is hierarchical; elements can have subordinate elements under them.

4. XML does not make any assumptions about the presentation mechanism. This is unlike HTML, which does make these assumptions.

2.5.2 Service Transport

Service Transport transfers messages between network applications and protocols are:

1. Hypertext Transfer Protocol (HTTP); a method used to transfer information on the WWW. The original purpose was to provide a way to publish and retrieve HTML pages [53];
2. Simple Mail Transfer Protocol (SMTP); a relatively simple process that transfers mail to another process on the same network or other networks through a gateway accessible to both networks [54, 55];
3. File Transfer Protocol (FTP); one of the oldest and widely used protocols on the Internet. It is used to connect two computers over the Internet so that the user of one computer can transfer files and perform file commands on another computer [56];
4. Blocks Extensible Exchange Protocol (BEEP); a framework for creating network application protocols. It is intended to abstract the common features that have traditionally been duplicated in each protocol implementation [57].

2.5.3 XML Messaging

This encodes messages in a common XML format, therefore messages can be understood at either end of a network connection. Today, this area includes protocols such as:

2.5.3.1. Simple Object Access Protocol (SOAP)

SOAP is the powerhouse of Web services. It is a specification for performing business method requests like XML applications. It uses the XML format and the HTTP protocol to enable any system to communicate on a socket over a simple protocol that can interoperate with any system. SOAP is a highly adaptable Object Oriented protocol that exists in over 80 implementations on every popular platform. It supports a flexible communication layer between applications, regardless of platform and location. The power behind SOAP lies in its simplicity. SOAP is a lightweight and very easy-to-understand technology, and is also easy to implement [58, 52].

In May 2000, W3C issued the specification for version 1.1 of the Simple Object Access Protocol [58]. SOAP initially meant Simple Object Access Protocol, but the name has been redefined to Service Oriented Access Protocol. The reasons for this are that SOAP is not simple; it is not object oriented as not all languages are object oriented. This flexibility in the protocol enables a program to be written in one language and run on one operating system and to communicate with a program written in another language that is running on a different operating system [15, 59].

The most important challenge in performing integration tasks using traditional middleware is the lack of a universal protocol. By being XML based and not tied to any specific language, SOAP has been developed to become the initial de facto standard protocol for performing integration tasks between many languages and on many platforms. A SOAP message is an ordinary XML document containing the following elements, as in Figure 2.5 [52, 60, 61]:

1. An Envelope element that defines a framework for describing what is in a message and how to process it;
2. An optional Header element that contains header data;
3. A required Body element that contains call and response data;

4. An optional Fault element that provides data about errors that occurred while processing the message.

SOAP is constructed on the messaging concept of passing XML documents from a sender to a receiver (endpoint). The XML document becomes known as a SOAP document and consists of three parts (Envelope, Header and Body). Figure 2.5 demonstrates the structure of a SOAP document [52, 60, 59].

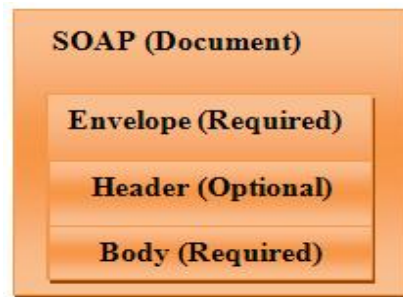


Figure 2.5: The structure of a SOAP document

SOAP is directly supported by the .NET, J2EE platforms and W3 standard. It is promoted by every major vendor in the industry, and with SOAP implementations working on virtually every platform, SOAP has established itself as the de facto standard for accomplishing Web services today. So, SOAP ensures that systems have a higher degree of interoperability with new schemes and with other organizations.

Here are some of the advantages of SOAP that has made it so popular, and there are more in Table 2.1. On the other hand, there are some disadvantages of SOAP, and we will also mention some of them, as follows [52, 62, 63, 59]:

Advantages of SOAP

- It is extensible and XML based.
- It is language and operating system independent.
- It can be used with multiple transport protocols and support by many vendors.

- SOAP can cross firewalls (security systems for protecting against illegal entry to a local area network that is connected to the Internet via computer).

Disadvantages of SOAP

- There are SOAP shortages of many advanced features. Many researchers and developers have written about the advantages of SOAP as a lightweight protocol, but in XML there are a host of missing features e.g. security.
- SOAP-based services are the foundation of the current push toward service-oriented Web architecture. Unfortunately, getting from the foundation to the complete edifice still involves a lot of work [58].

2.5.3.2. XML-RPC

XML-RPC is a set of implementations that enables software to run on different operating systems and in different environments to make it easy for computers to share resources and procedure calls across the Web. Therefore, users are able to directly access to the resources that they need to process, not only to read and reuse systems. It is designed to be as simple as possible, while letting complex data structures to be transmitted processed and come back as in Figure 2.6 [64, 65, 60]..

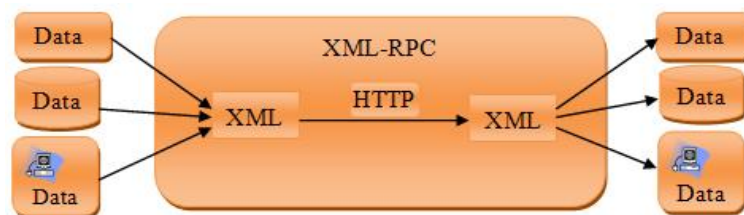


Figure 2.6: XML-RPC Structure

2.5.3.3. SOAP vs. XML-RPC

Commonly, everything that SOAP can do, XML-RPC can as well. XML-RPC is considered to have been widely adopted for a few years longer than SOAP. This means that XML-RPC toolkits are probably more stable and mature. XML-RPC's simple specification has also contributed to its wide adoption and success, and indicates that the protocol will probably be used for a few more years to come despite SOAP's entrance into the WWW trends. Table 2.1 shows the differences between SOAP and XML-RPC [66, 62, 63, 60, 59].

Table 2.1: SOAP vs. XML-RPC

No	SOAP	XML-RPC
1	In general, everything SOAP can do...	XML-RPC can as well.
2	It appeared after the XML-RPC language.	It is considered to have been widely adopted for a few years longer than SOAP.
3	It is complex but more extensible, and has better support for complex data expressiveness.	XML-RPC's simple specification has also contributed to its wide adoption and success, and indicates that the protocol will probably be used for a few more years to come despite SOAP's entrance into the computing world.
4	The concept of SOAP is more complex than that of XML-RPC.	It has a very simple XML vocabulary, and this makes it easy for vendors to implement the specifications.
5	It makes heavy use of XML namespaces and XML Schema; the standard allows developers to construct Web services of any kind, providing complete customization and control.	It is difficult to create some XML-RPC services such as trade agreements and other document-literal services because of the absence of intermediates and document-exchange features.
6	It is known to be more scalable with larger documents as document-style SOAP is more efficient.	XML-RPC specification defines a limited number of supported data types.
7	It has established itself as the de facto standard for accomplishing Web services today.	-
8	It is directly supported by the .NET, J2EE platforms and W3 standard.	XML-RPC proprietary and frozen.
9	It has XML Namespaces, and supports XML Schema.	It does not have any XML Namespaces, does not support XML Schema and has a very definite set of types and primitives.

2.5.4 Service Description

This is used for describing the public interface to a specific Web service. The WSDL interface format is typically used for this purpose.

2.5.4.1. Web Services Description language (WSDL)

WSDL is an XML-based format published for describing Web services that are implemented by using SOAP, and made accessible from the UDDI directory [22]. The browser who is wishing to access Web services can read and interpret the WSDL file to discover the location of the service and the available operations. In this situation, WSDL can be defined as the initial Web service interface that gives the browser the information that he/she needs to interact with the service [5, 12, 17].

WSDL describes Web service acts as a contract between the Web service user and the server. In this contact, the service provider and requester are able to exchange data in a standard way [67]. One of the most important advantages of the WSDL format is that it allows dividing the abstract functionality description offered by a service from the concrete details description, such as message format and communication protocol (HTTP or SOAP) [68].

WSDL file is a Web service using some main elements. These elements have been considered stereotyped classes because they represent important components and are explicitly defined in WSDL [67, 68, 6]:

2.5.4.1.1. Types

This encloses data type definitions that are relevant for the exchanged and sending messages between client and server. Figure 2.7 displays the structure of type, which is part name, element and type.

Recently, XML Schema Definition (XSD) has become ever more important when


```

<definitions....>
  <types>
    <part name="id" element="xsd:string" type="xsd:int" />
  </types>
</definitions>

```

Figure 2.7: The Structure of Type Element in WSDL

it comes to conceptualizing knowledge and to define programming language independent type systems. An XML Schema instance is an XSD and it has the filename extension ‘.xsd’. The XSD type system can be used to define the types in a message whether or not the resulting wire format is actually XML, or whether or not the resulting XSD schema validates the particular wire format. This is exciting if there are multiple bindings for the same message, or even only one binding, but that binding type does not already have a type system in common use [67, 69, 40].

2.5.4.1.2. Messages

Message describes the names and format of the messages supported by the service. Message contains one or more logical parts and each one is associated with a type from some type system by using a message-typing attribute. The syntax of message is shown in Figure 9.6:

```

<definitions....>
  <message name="getCoursesRequest">
    <part name="id" element="xsd:string" type="xsd:int" />
  </message>
</definitions>

```

Annotations:

- It supports a unique name between all messages defined inside the including WSDL file. (points to **getCoursesRequest**)
- It refers to an XSD element using a QName (points to **xsd:string**)
- It support a unique name between all the pieces of the including message. (points to **id**)
- It support a unique name between all the pieces of the including message. (points to **xsd:int**)

Figure 2.8: The Structure of Message Element in WSDL

The message-typing attributes are in bold as in the middle of the above message. The message name attribute supports a unique name between all messages defined

inside the including WSDL file.

2.5.4.1.3. Port Types

Using the definition from WSDL documentation, a port type is a named set of abstract operations and the abstract messages involved. They are a collection of operations that describe the operations performed by the service through the defined interface. The syntax of port type is shown in Figure 2.9:

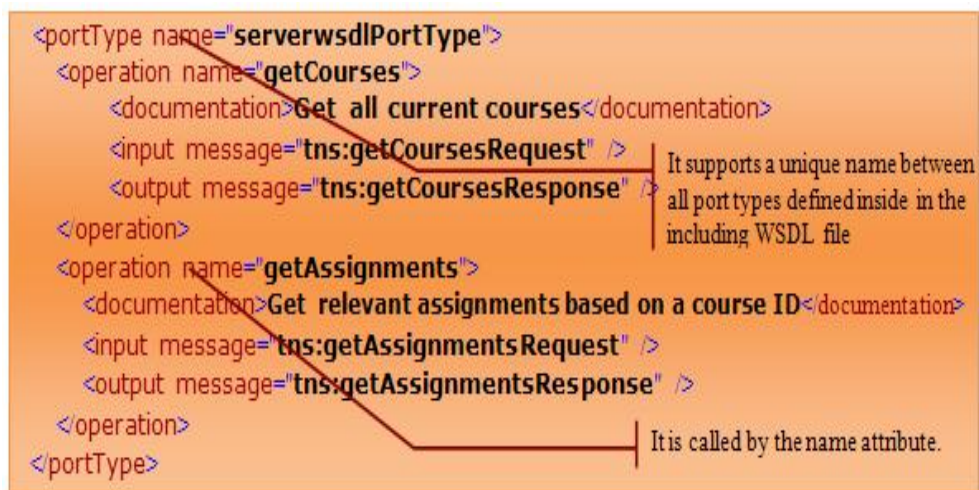


Figure 2.9: The Structure of Port Type Element in WSDL

As we can see in Figure 2.9, a port type element has a unique name that identifies it from any other port type elements. It includes any number of child operation elements and these operations exist for each operation that will be exposed by the Web service named during its name attribute. When writing a WSDL document, the value of the name attribute is the name of the corresponding PHP function.

2.5.4.1.4. Binding

This defines the message format and protocol details for messages and operations defined by a particular port type. The grammar for a binding is as follows:

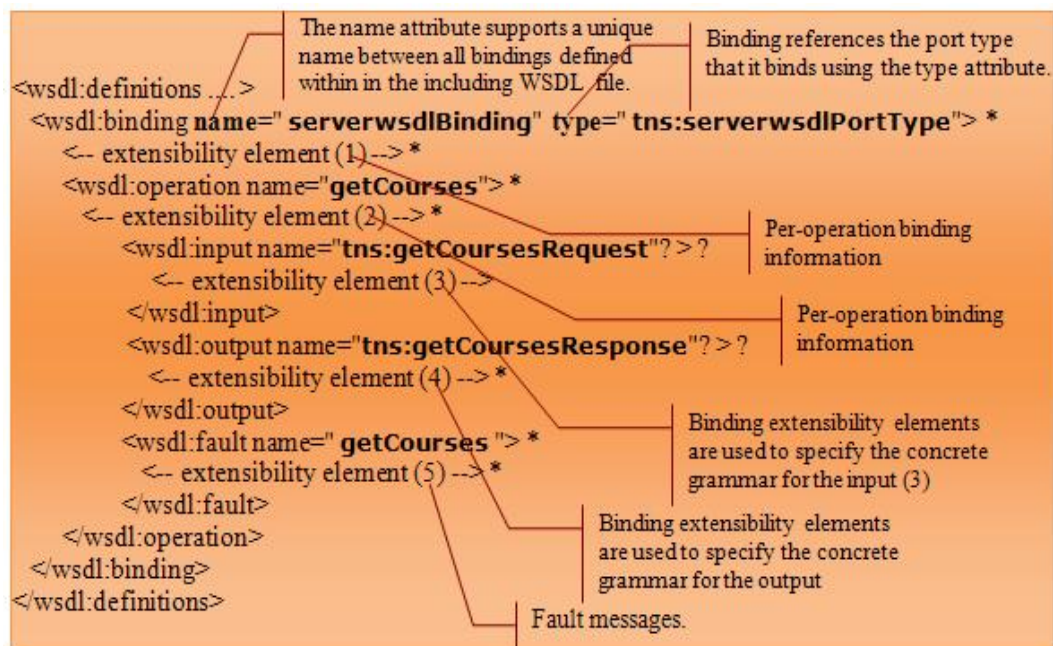


Figure 2.10: The Structure of Binding Element in WSDL

The binding is between a port type and both network protocol and message format, and defines the service interface in a concrete way. It must identify accurately one protocol and not specify address information. When defining bindings, each binding must have a unique name, identified by the value of the name of attribute. The value of the type attribute identifies the port type for which this binding is used. Using the port type described in the previous part, which was named `serverwsdlPortType`, and it can begin constructing the binding as in Figure 2.10.

2.5.4.1.5. Ports

These specify an address for a binding and define an individual endpoint by specifying a single address for the binding. The syntax of port is shown in Figure 2.11:

The port does not have to specify more than one address and does not have to specify any binding information other than address information.

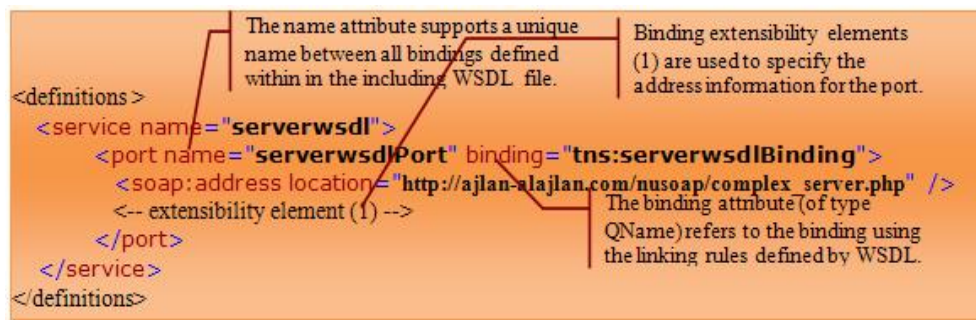


Figure 2.11: The Structure of Ports Element in WSDL

2.5.4.1.6. Service

This defines one or more port and specifies the address (URL) for accessing the service. The syntax of service is shown in Figure 2.12:

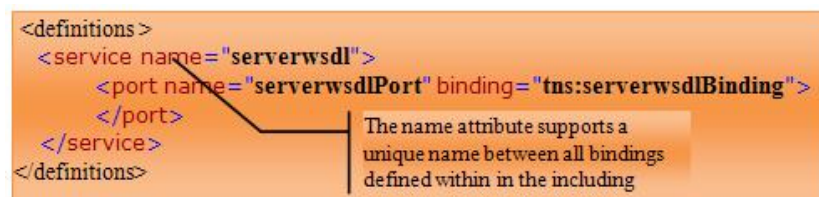


Figure 2.12: The Structure of Service Element in WSDL

The relationship between ports and service contains the following points:

- The outcome of one port is not the input of another.
- If the services have a number of ports that divide a port type, but employ different bindings, the ports are alternatives. Every port supports semantically equivalent behaviour. It lets the user of a WSDL document choose a port to meet a particular description to communicate with, based on some criteria.
- We can determine a service's port types. It lets the user of a WSDL document determine whether or not to communicate with a specific service, based on whether or not it provides a number of port types. It is helpful if there is some implied relationship among the operations of the port types, as the entire set of port types have to be present in order to achieve a specific assignment.

1.5.4.1.7. Operation

An operation is an abstract description of an action supported by the service as in Figure 2.13.

```
<operation name="courseDetails">
  <soap:operation soapAction="serverwsdl#courseDetails" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="serverwsdl"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="serverwsdl"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
```

Figure 2.13: The Structure of Operation Element in WSDL

With WSDL, SOAP is much easier to deal with, both for users and developers. WSDL is used to describe SOAP over HTTP and it has a number of XML elements, as in Table 2.2, which summarizes those elements and their attributes and children [70]:

Table 2.2: XML elements of WSDL

No	Element	Attribute	Children
1	<definitions>	Name Target Namespace xmlns (other namespaces)	<types> <message><portType> <binding><service>
2	<types>	(none)	<xsd:schema>
3	<message>	Name	<part>
4	<portType>	Name	<operation>
5	<binding>	Name type	<operation>
6	<service>	Name	<port>
7	<part>	Name type	(empty)
8	<operation>	Name parameterOrder	<input> <output> <fault>
9	<input>	Name message	(empty)
10	<output>	Name message	(empty)
11	<fault>	Name message	(empty)
12	<port>	Name binding	<soap:address>

2.5.5 Service Discovery

This compacts services into a common registry as the network Web services can publish their location and description, and it makes it easy to discover what services are available on the network. At present, UDDI is used for service discovery [32, 47].

2.5.5.1 Universal Description Discovery and Integration (UDDI)

Universal Description, Discovery, and Integration (UDDI) project is a sweeping industry initiative. It creates a platform-independent, open framework for describing services, discovering businesses, and integrating business services using the Web [61]. It has three components [15, 71]:

- White pages, which contain address, contact details, and known identifiers for Web services providers.
- Yellow pages that provide industrial categorization of Web services based on standard taxonomies.
- Green pages that contain technical information about services.

UDDI provides a mechanism for customers to find Web services. Using a UDDI interface, businesses can dynamically look up and discover services provided by external business partners. A UDDI registry has two kinds of customers:

- Businesses that want to publish a service description (and its usage interfaces).
- Customers who want to obtain services descriptions of a certain kind and bind programmatically to them (using SOAP).

UDDI itself is layered over SOAP and assumes that requests and responses are UDDI objects sent around as SOAP [5]. UDDI is comparatively light-weight and contains enough information to direct users to resources hosted outside it [71].

2.6 What are Web Services Good for?

Web services are not replacements for any technologies but rather are complementary to other tools in the toolbox. They represent loosely coupled interactions, which are better fitted to integrating disparate software domains and bridging incongruous technologies rather than heavy-duty and higher-performance applications. In addition, they are excellent for submitting documents to long-running business process flow, which seems in any case to be a good approach to begin with when performing integration over the Web [14].

Many analysts believe that Web services will become a multi-billion-dollar market in the future. They also realize that companies loath crafting SOAs to support these models. Organizations will not fully embrace the model without proof that the Web services they would like to exchange will be supported by most or all of the architectures of their suppliers, customers and partners.

W3C recently received a submission for the formal approval of WS-Policy as an industry standard, after its developers had filled a hole in the specifications that made companies reluctant to write to it. WS-Policy was introduced in 2002 by IBM, Microsoft, BEA Systems and others as a means for Web services to express their requirements and policies to other Web services [72].

2.7 Who is Using Web Services?

The best example of the growth of Web services is auction engines such as the eBay marketplace. The eBay Web site offers a highly successful auction service. It has been developing Web services platforms by extending application programming interfaces, which essentially turn a Web site into a platform. Tim O'Reilly has said in a White Paper for eBay's developer section [73]:

'We are moving from a world in which the software that most people

interact with is desktop software residing on a personal computer into one in which software is running as a set of services provided over the Internet' [73, pp.4].

The auction site's developer section provides wide-ranging information about deploying an eBay Application Programming Interface (API). Since 1999, eBay has offered APIs and currently offers more than one hundred Web services calls available to developers to build applications that can connect to those services. They contain payment options, pricing information, and buy-it-now features through the PayPal subsidiary. The growth and use of APIs across the Web exemplify how quickly Web services are spreading, even as technical issues such as security and authentication are worked out by standards bodies.

Another example is Amazon.com. Some companies, such as Microsoft and Sun Microsystems, have been helping its developers build and deploy Web services and clients for Amazon.com for close to four years now [32].

2.8 The Future of Web Services

The future of Web services is related with the future of Web because Tim Lee, who invented the WWW, has said that the next stage of the Web will be around data not text (HTML (text), XML (data)). Many of the more significant aspects of the distributed system are not available yet, for example security, routing, transaction-setc. These features will be rolled out over time with standard SOAP headers. The unusual growth of the WWW has given developers and researchers the ability to easily and cheaply distribute electronic documents to international audiences [14].

The next generation of the Web is addressing some of the disadvantages of the current Web, and is trying to find the best solutions for the more difficult Web searches e.g. exact matches on text strings embedded in Web pages. The future

of Web has to be achieved as an extension and evolution of the existing Web, as it is not possible to substitute the entire thing and begin again. The solutions for application-to-application communication difficulties have to be taken from Internet.

Microsoft is energetically working on a group of higher-level services known as Global XML Web services Architecture (GXA). GXA is a set of technologies that aims to make Web services appropriate for application integration across platforms and over the Internet. Most GXA specifications are proposals for SOAP headers, which provide other features of the enigma. Web services developers are intending to aim the enormous global network of the Web, established for human interaction, at entirely new goals. Software-oriented interactions will automatically perform processes that beforehand required manual intervention. For example searching and buying goods and services at the best price; streamlining business procurement, billing, and shipping operations; and organizing travel tickets and restaurant timetables for given information [14, 74, 75].

The Web services of the future will be truly providing information, content, and updates targeted to specific customers. Anytime and anywhere service will be possible, delivered on platforms ranging from wireless to traditional PCs and notebooks. Additional interfaces such as Wireless Application Protocol (WAP) for lower bandwidth wireless connection will be used and maximized. All elements of corporate and customer knowledge bases will be linked and leveraged to permit organizations to complete service profile of the customer in every interaction [75].

Web services are part of the Web's future and they are being developed by the largest computer corporations almost entirely in terms of standards bodies. So, in addition to everything else, Web services developers must learn some basic familiarity with the major players because if they do not understand the institutional context of the relevant standards, they will find it hard to enact or to judge standards compliance [76].

2.9 Summary

This chapter has presented the key point about SOC, in that it involves extended, loosely coupled activities among two or more independent business partners. Such activities can be thought of as business processes that engage several services in a manner that brings about the desired business outcome. Adopting SOC has the potential to bring about decreased programming difficulty and expenses, lower costs, quicker time-to-market, new revenue streams and improved operational efficiency.

In addition, this chapter has presented a survey of Web services, which are fast becoming an important technology in the evolution of distributed computing over the Web. Their technologies are rapidly changing, and a long list of additional features and functionality is required to complete the vision. Web services allow applications to be integrated faster, more easily and more cheaply than ever before, and they can influence the data independence of XML to solve EAI problems, both outside and inside the firewall. The basic Web services standards SOAP, WSDL, and UDDI are immediately useful for many applications, such as publishing interfaces to automated business processes, bridging disparate software domains, and connecting wireless customers for Web purposes.

The next chapter presents the key points about VLEs, which are a form of e-learning software packages that enable online interactions of various kinds to take place between teachers and students. It presents an outline of the literature on VLEs and their benefits. The architecture, challenge and functional and non-functional requirements of VLEs are also described in this chapter, as well as the main aspects of the evaluation of VLEs. The main features and tools of VLEs are described in detail together with the benefits, criteria and limitations of OSSs generally, and why they are used. The Future of VLEs is mentioned in the end of this chapter.

Chapter 3

Virtual Learning Environments as E-learning Systems

Objectives

- To provide an overview of the literature on VLEs and its benefits.
- To focus on architecture and challenges of VLEs.
- To describe the main aspects, features and tools, and the future of VLEs.
- To give an overview on the benefits, criteria and limitations of OSSs.

3.1 Introduction

Using the Internet to enhance e-learning has become a trend in modern higher education institutes. E-learning platforms are increasingly becoming a significant part of the strategy for delivering online and flexible e-learning [77]. At present, an explosion is occurring in the demand for e-learning all over the world. The next

generation of e-learning needs to provide greater dynamism and flexibility to support today's increasingly pressurized e-learning requirements [78, 79].

E-learning systems are a group effort, where educators, administrators, and all users from a variety of other areas of expertise come together in order to serve a community of learners [80, 81]. E-learning offers institutes a number of benefits, such as access anytime and anywhere. VLEs are a form of e-learning software that enable online interactions of various kinds to take place between teachers and students. In order for current and future generations of personalized VLEs to improve learning efficiency and effectiveness, there are essential requirements that have to be realized. In the last decade, a number of VLEs have been adopted and developed by universities [82, 83, 84]. There are more than 250 e-learning packages and more than 45 of them are Open Source Software (OSS) as in Table Table 3.4 [82, 85, 86].

This chapter presents an outline of the literature on VLEs and their benefits. The architecture, challenges and the main aspects of evaluating VLEs are also described on this chapter. The main features and tools of VLEs are described in more detail in this chapter, together with the benefits and limitations of OSSs and why they are used. The future of VLEs is mentioned at the end of this chapter.

3.2 Virtual Learning Environments

Nowadays, many researchers and developers have tried to use the emerging computer and communication technologies to establish effective learning and teaching environments [87]. VLEs offer a number of benefits for institutes such as anytime, anywhere access, improved motivation, better integration of information and communication technology tools, opportunities for independent learning, and increased parental engagement [80, 85, 88].

Production service VLE systems are a form of e-learning software. They are a

set of integrated learning tools designed to support a student's learning experience [79]. The principal tools of a VLE package contain online support for teachers and students, curriculum mapping, Internet links to outside curriculum resources, student tracking, and electronic communication [82, 88].

3.2.1 The Definition of Virtual Learning Environments

VLEs are computer applications that facilitate online teaching and learning. They are e-learning systems that support a range of learning contexts, ranging from conventional, classroom implementation to off-line or online learning, and distance learning [88]. VLEs are essentially websites that provide various basic functions deemed valuable in the learning process. Usually, a number of tools and navigation aids are provided, with the aim of placing any online educational materials into a clear, organised structure. The environment provides students with easy access to online courseware, questionnaires, communication tools, course documents, and lecture notes [80, 89].

VLEs are a set of teaching and learning tools designed to support a student's learning experience by including computers and the Internet in the learning process. The scenario of VLE software is that it runs on a server which can be accessed from anywhere with an Internet connection. The Europe Wide has defined VLE, and this definition seems to reflect a perception that VLEs can be [85]:

'Facilitators of changes in education and pedagogy towards more learner centred approaches, enhancing interactivity in learning [and] helping constructional knowledge building'.

The Joint Information Systems Committee has also defined VLEs, and this definition seems the most widely accepted [85]:

'A VLE is an electronic system that can provide online interactions of

various kinds that can take place between learners and tutors, including online learning' [85, pp.8].

In the beginning, VLE software packages were seen as primarily of interest to the area of e-learning but they are now seen as having a wide range of applications in traditional institutes. This view has been shaped by factors such as [89],

1. The growing number of staff and students with access to the Internet.
2. Learning, teaching and quality issues.
3. The desire to make more effective use of IT infrastructure in universities.
4. The needs and challenges of teaching more students in larger groups.

This is echoed by [90], which has defined VLEs as follows:

'VLEs are learning management software systems that synthesise the functionality of computer-mediated communications software (e-mail, bulletin boards, newsgroups etc) and on-line methods of delivering course materials (e.g. the WWW). To date, several different packages have appeared from both leading commercial vendors and university-based projects. Other systems are currently under development' [90, pp.3].

VLEs are increasingly seen as important tools by higher education institutes, as VLEs contain many components for both learners and educators, who can then participate in a wide variety of online interactions, particularly but not exclusively online learning [85, 86]. Most British Universities will be using VLEs to support at least some of their teaching and learning within the next few years. Qassim University in Saudi Arabia therefore needs to develop its expertise in this area for competitive reasons as well as any other perceived benefits. VLEs are also seen as increasingly useful for collaborative projects, even with other universities [80, 89].

3.2.2 The History of Virtual Learning Environments

Following the emergence of the Internet in the early 1990s, many systems have been developed to exploit its benefits. Since the middle of 1990s, educators have witnessed the appearance of e-learning systems such as VLEs whose purpose it is to provide tools in order to support teaching and learning across the Internet. In the last decade, a number of VLEs have been adopted and developed by universities; many educators want to take this opportunity to capitalise on the benefits offered by the Internet to support their own teaching tools [82].

VLEs are a new development and there is therefore not a great deal of history about them. In recent years, they have been becoming increasingly popular, and the introduction of several online learning companies has helped with this steady progression of VLEs. Nowadays, VLEs are quite popular amongst universities and the Open University is a perfect example; its system is named Moodle. At the end of 1980's and towards the middle of 1990's, educational technology developments in the UK developed from Open Learning, through to Computer Assisted Learning (CAL), which included software and hypertext on CD-ROM and file servers [91].

3.2.3 The Benefits of Virtual Learning Environments

As with any technology used in teaching and learning, VLEs have no intrinsic educational value in themselves. The way in which online courses and activities are designed and delivered can add value and increase effectiveness. The main reason why VLEs have become so popular and embedded in many institutions is that because there are actual and quantifiable benefits to be gained from the use of this technology. Ever increasing student numbers is one obvious aspect of Higher Education where VLEs can help. They can maintain good communication and there are opportunities for automated assessment. In terms of widening participation, VLEs can support and offer resources to, for example, part time students who cannot

travel to the institute campus all the time. Below are some commonly perceived advantages and disadvantages of using VLEs [80, 88, 92, 93].

Advantages

1. There are many advantages of using VLEs for the learner and indeed for the institution.
 - Tutors can use them to manage courses such as tracking student progress, making announcements, issuing timetable information, setting, receiving and marking assignments, creating multiple choice tests, and so on.
 - They also offer huge advantages to the universities and their staff, as they allow them to upload files onto the system for everyone to access; this saves them having to attach documents to e-mails and then sending them to the class.
2. Both the educator and the student can enjoy the privacy of their home environment. The Internet provides cheap and easy access to information sources of huge diversity. Interactivity is offered on a large scale and variety, and this technology even provides drills and exercises for basic skills [77].
3. VLEs enable interaction between students and instructors almost free of time and location constraints [88].
4. VLEs enable the benefits of integrating individual and group learning facilities, and Web services will offer this for VLE systems.
5. Easy online delivery of materials and easy to use for students and lecturers.
6. Offers flexible support for educators who do not need to be in a fixed time or place to support and communicate with students.

7. Has the potential for new ways of learning and teaching such as active and independent learning, which make use of online communication, online assessment and collaborative learning.
8. Makes education available to the wider population [88].
9. VLEs are used for a whole host of things that both teachers and students need:
 - Manage studying by using the online calendar.
 - Students can submit their work and know when the deadline is.
 - If students miss a lecture, they can get the slides on the VLE.
 - Contact important people such as admin, lecturers, instructor, etc.

Disadvantages

Although VLEs offer many advantages for teachers, students and institutes, they also have some limitations, and we will mention some of them as follows:

1. Both teachers and learners need training in order to be able to use a VLE. As with every new technology, it takes some time to learn and gain confidence.
2. VLEs can be a dumping ground for material not designed for delivery online.
3. Copyright, legislation and accessibility of materials need to be considered.
4. Online support must be carefully planned and learning materials can become outdated.
5. VLEs are expected to become more sophisticated and this requires more staffs.
6. Hundreds of articles, long lists of links, and high-resolution pictures and videos all delay accessibility and lower satisfaction levels.

3.2.4 The Architecture of Virtual Learning Environments

There is no consistent stranded architecture for VLE systems. All the available architectures depend on the kind of VLE product and the institute's requirements. Therefore, there are no particularly significant differences between them and every product has similar tools, and activities. The differences lie in the institute's requirements and some universities integrate one or more VLE product with each other, such as Oxford University which has integrated two OSS learning environments, Bodington VLE and Moodle [94], which are slightly different from each other.

In this section, we will mention two VLE architectures as examples used by some universities. As shown in Figure 3.1, VLEs act as a gateway to online learning, to support features, and to technical specifications tools, and these are shown below. The diagram is based on the one circulated widely in Autumn 1999 by BECTa, and its main purpose is to position VLE as a sub-system within the range of information systems and processes that a college's e-learning environment includes [95].

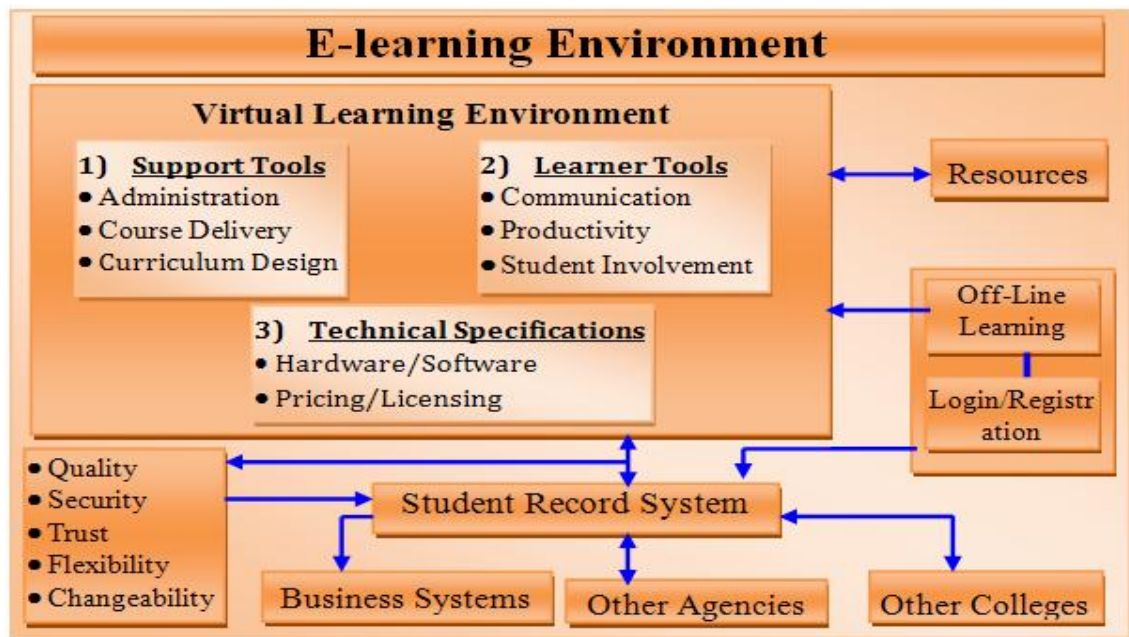


Figure 3.1: VLE as a Sub-System within the E-learning Environment

As in [1], the aim is to build particularly multi-user VLEs using the understanding gained from prior efforts in distributed information systems. Users experience a direct view of a locale within much larger and contiguous VLEs, as in Figure 3.2.

Users are able to move in a continuous environment from one position to the next. The VLE functions as hyperlinks to web pages and other Internet-deliverable services that can be viewed via the client embedded in a Croquet frame. Authenticated Users would be able to see and interact with each other while their client caches data about a locale and renders the scene in real-time.

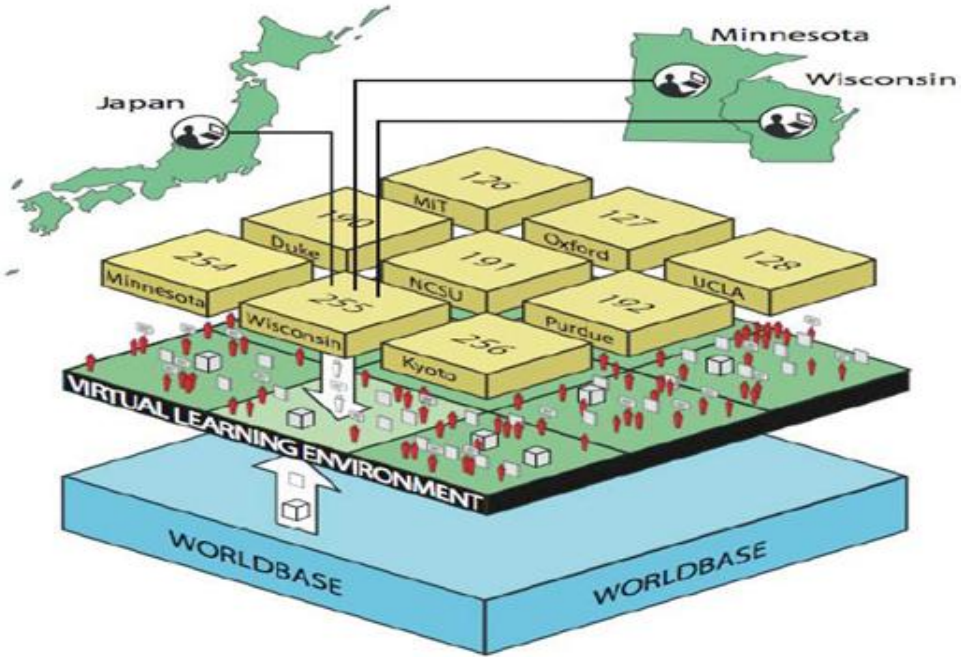


Figure 3.2: User-experience Architecture [1]

Interactivity servers are fundamentally long-lived Croquet clients, and admit connections from all clients for a VLE position to maintain transient or non-cacheable objects. Since the Interactivity server goes back up the list of users in their positions to new visitors, it has been accepted as the norm for authenticating a user's identity and supporting gathering credentials. The Worldbase servers distribute digitally signed, cacheable, authoritative data objects describing long-term persistent

aspects of any position within the VLE, and a network locator for each position's Interactivity server.

3.2.5 Functional and Non-Functional Requirements of Virtual Learning Environments

Whilst an initial investment of time is needed to establish and configure VLEs, the implementation of the system should reduce administrative workload and free up teachers' time for teaching. Table 3.1 displays some functional and non-functional requirements of VLEs [89].

Table 3.1: Functional and Non-Functional Requirements of Virtual Learning Environments

Functional Requirements	Non-Functional Requirements
<ul style="list-style-type: none"> ➤ VLEs should be compatible with relevant IT systems of the host institution both at a departmental and administrative level. ➤ The user interfaces should be simple, intuitive and easy to use. ➤ VLEs should provide opportunities to improve the quality and variety of teaching and learning that are not being achieved using current methods. ➤ In general, VLEs must offer the educator a dynamic way of teaching a particular subject using the Internet. 	<ul style="list-style-type: none"> ➤ VLEs usability for both learners and educators. ➤ Agility, ease of change for future improvements. ➤ They should realize a community based learner-centered pedagogy.

3.2.6 The Main Aspects of Evaluation for Virtual Learning Environments

It is important to evaluate the existing VLE systems in order to select the most suitable one that meets the future needs (requirements and specifications) of an academic institution. The next chapter explores this in greater depth, evaluating Moodle against some other well-known VLE products, but here we outline and four carefully selected criteria, which will be used to make that evaluation, as in Table 3.2 [89, 90].

Table 3.2: The Main Aspects for Evaluating Virtual Learning Environments

No	Criteria	Evaluation
1	Functionality	<ul style="list-style-type: none"> ➤ Does the system support a range of browsers/platforms? ➤ Can you import pre-existing materials? ➤ Can tutors add resources, including non-textual resources? ➤ Can the VLE be accessed from different computer platforms?
2	Management/ assessment	<ul style="list-style-type: none"> ➤ Can the system store and view data about learners? ➤ Can the tutor create assignments/tests? ➤ Can the tutor assess assignments/test? ➤ Can students create/import content?
3	Flexibility/ pedagogy	<ul style="list-style-type: none"> ➤ Can data be exported to University admin systems? ➤ Can assignments be submitted via the system? ➤ How easily can a course be modified after commencement? ➤ How easy is it to design a course?
4	Accessibility	<ul style="list-style-type: none"> ➤ Are accessibility design tools inbuilt? ➤ How easy is it for the tutor to follow accessibility guidelines? ➤ Are there special hardware/software requirements? ➤ Does the product address issues that relate to KSA?

3.2.7 The Challenge of Virtual Learning Environments

Using the Internet to support teaching or learning has become increasingly popular in education institutions all over the world. However, the majority of VLEs face some challenges and they are still limited to the dissemination of teaching materials. Neither have the strengths of the Internet been fully exploited nor have the functions been fully utilized, for instance, supporting autonomous, exploratory, interactive and collaborative learning. Nevertheless, many developers have attempted to use the emerging computer and communication technologies to create effective learning and teaching environments [87].

The next generation of VLEs could address the limitations of the current on-line systems by providing a richer social context for online learning [96]. Existing e-learning systems have failed to provide support for the deeper interaction and collaboration necessary to realize a community based learner-centred pedagogy. These current systems have also failed to fully benefit from the powerful graphics and simulation capabilities of modern computers. Current e-learning systems are designed primarily as centralized server architectures and have scalability problems that limit their use for large-scale cross-institution collaboration [1, 96].

Another challenge that faces VLEs is to identify and maximize the pedagogical possibilities that distinguish it from traditional classroom learning. This is the challenge of using VLEs for highly productive e-learning and avoiding the temptation of merely replicating the traditional classroom, and instead, of focusing on realizing the teaching and learning experiences that VLEs invite and make possible. Like distance learning itself, VLEs are neither a substitute for nor a complete departure from the traditional classroom, but a compelling extension and development of that architecture and its pedagogies.

As in [97], VLEs try to capitalise on two existing active areas of development. The first is an SMS text messaging extension, which is used to support personal-

ized messaging. The second is a Bluetooth-based communications service, named BlueZone, which is used to complement SMS text messaging in order to present an alternative communications platform to learners. The combination of these technologies supports a unified and charge-effective delivery mechanism for communicating with university undergraduates on a large scale.

VLEs should not be designed in a vacuum, rather they should match students' needs and desires as closely as possible, and be adaptable during course progression. Therefore, as in [90, 97], there are a number of challenges within higher education that VLE developers should consider:

1. Increasing student numbers.
2. Automated assessment.
3. Widening participation.
4. Improved access to limited resources.

With enormous quantities of diverse learning materials, teachers and students still feel that retrieving and integrating educationally relevant learning materials in VLEs is a challenging assignment. Two significant matters that are often raised are how to increase the degree of relevancy among retrieved learning materials and how to deal with the heterogeneity of varying data sources [80].

3.3 The Main Features and Tools of Virtual Learning Environments

VLEs enable teachers to build resources fast and without the need to develop technical skills. They provide an integrated set of Internet tools, which allow the fast upload of materials and offer a consistent look and feel that can be customised by the user. VLE tools are criteria-based, and they enable developers to evaluate and select the most suitable VLE product. No single product can possibly meet all these criteria, and the most suitable within a specific context may not be perfect for interface, technical, functional, or cost reasons. Table 3.3 describes the tools and features that should be considered if a VLE is to facilitate a complete learning and teaching experience [98, 87, 90, 93, 99].

Table 3.3: The Main Features and Tools of Virtual Learning Environments

The Main Features and Tools of Virtual Learning Environments		
1. <u>Communication Tools</u> 1. Discussion Forums 2. File Exchange 3. Internal Email 4. Online Journal/Notes 5. Chat 6. Video Services 7. Whiteboard	4. <u>Curriculum Design</u> 1. Accessibility Compliance 2. Course Templates 3. Curriculum Management 4. Customized Look and Feel 5. Instructional Standards Compliance 6. <u>Instructional Design Tools</u> 7. Content Sharing/Reuse	7. <u>Hardware/Software</u> 1. Client Browser Required 2. Database Requirements 3. Server Software 4. UNIX Server 5. Windows Server
2. <u>Productivity Tools</u> 1. Bookmarks 2. Orientation/Help 3. Searching Within Course 4. Calendar/Progress Review 5. Work Offline/Synchronize	5. <u>Course Delivery Tools</u> 1. Course Management 2. Instructor Helpdesk 3. Online Grading Tools 4. Student Tracking 5. Automated Testing and Scoring	8. <u>Pricing/Licensing</u> 1. Company Profile 2. Costs 3. Open Source 4. Optional Extras 5. Software Version
3. <u>Student Involvement Tools</u> 1. Groupwork 2. Self-assessment 3. Student Community Building 4. Student Portfolios	6. <u>Administration Tools</u> 1. Authentication 2. Course Authorization 3. Registration Integration 4. Hosted Services	

3.4 Open Source Softwares

OSS is the acronym for Open Source Software, which is becoming widely adopted by educational institutions. Programmers use source codes as a language to create software and these codes are open source, meaning that they are available for everyone. OSS uses source codes that are unrestricted and freely available by downloading from the Internet. There are hundreds of OSSs already available, ranging from simple email software to WWW servers (Apache) and full operating systems (Linux). OSS is an emerging issue for all education and training sectors across the world [79, 2, 100].

3.4.1. What is Open Source?

There is increasing interest in the existing applications of OSS in institutes in both the public and the private sectors. OSS is defined as a licensing agreement that requires the source code to be available and distributed with the program itself. Depending on the exact license, users are unable to take the code and make it a part of a closed source software package. In closed source, users get a working program, but they are not able to access the code and see how it is written or make any modifications. In contrast, OSS means that users are free to download, use, modify and even distribute it under the terms of GNU [79, 100]. OSS has essential features such as free re-distribution, available as code for everyone and modification or derivation of other software from it [100].

3.4.2. Open Source Server

The structure of an open source server is as in Figure 3.3. This server provides open online communities for educators and learners [2].

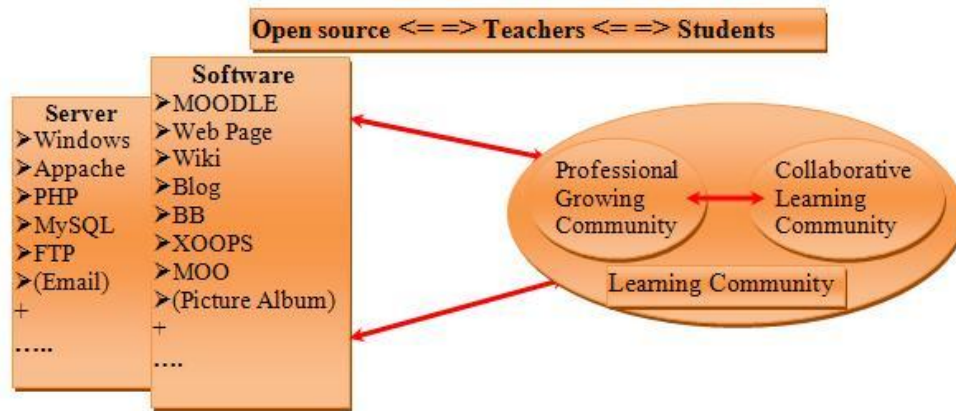


Figure 3.3: Structure of Open Source Platform [2]

3.4.3. The Criteria of Open Source Software Packages

Open source does not just mean access to the source code. The distribution terms of OSS have to comply with the following criteria [101, 102, 103]:

1. Source Code

All VLE platforms have source codes, but some of these are OSS codes, which allows developers to modify and distribute them under license. Deliberately encrypted or confused source code (closed) is not enabling because programs cannot be developed or modified. In OSS, the aims are to make evolution and modification easy.

2. Integrity of the Author's Source Code

The license can limit OSS from being distributed in modified form only if the license enables the distribution of patch files with the code for the aim of modifying the program at build time. This license should be clear to authorize developers to work and distribute the source code.

3. No Discrimination against Persons or Groups

The license should not distinguish against any group and person. In order to get the maximum advantage from the process, the maximum variety of persons and groups should be equally qualified to contribute to open source. So we prohibit any open source license that locks anybody out of the process.

4. Distribution of License

The rights attached to the program have to apply to all to whom the program is redistributed without the need for execution of an additional license by those parties. This clause is intended to ban closing up software by indirect means such as requiring a non-disclosure agreement.

5. Free Redistribution

The license should not limit any party from selling or giving away the software as a component of a total software distribution containing programs from a number of different sources. It should not require a royalty or other charge for such sale.

6. No Discrimination against Fields of Endeavour

The license must not limit anybody from making use of the program in any area of endeavour. For example, it may not limit the program from being used in business, or from being used for genetic research.

7. License Must Not Restrict Other Software

The license should not put limitations on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software. Distributors of OSS make their own choices about their own software.

3.4.4. The Limitations of Open Source Software

The use of OSS is, for most organisations, the most effective way to reduce costs and improve system reliability and security. With OSS, institutes can take control of their computer resources and manage their IT future. Nevertheless, there are several disadvantages for OSS, as follows [100, 102, 2]:

1. OSS is against the principles of commerce.
2. It is against human nature to work for nothing. Development is usually only done by hackers and students.
3. Restricted choice; in every area of software, there are many of choices for different commercial packages, but rarely are there more than one or two open-source options.
4. Nobody controls development. Anybody can change the software, which eventually becomes unstable and insecure.
5. There is no one to turn to for support.
6. When the lead developer leaves, the project dies.
7. Open-source projects eventually splinter, just like Unix.⁷⁶

3.4.5. Indicative List of Open Source of E-learning Systems

There are more than 250 platforms of e-learning and more than 45 of them are OSS, as in Table 3.4. The best known of these are Moodle, Ilias, Eduplone, Claroline, SAKAI, WebCT and Bscw, and they have wide developer communities and present robust arguments for considering open source as a straight and potentially feasible competitor to commercial products. One OSS of particular note, which has emerged to meet the growing interest in open-source platforms, is Moodle [86, 98, 104].

Table 3.4: Indicative List of Open Source Management Systems

No	Name of Product	Author/Sponsor/developer	Country	License
1	ATutor 1.2	University of Toronto	Canada	GPL
2	Adept	http://adept.sourceforge.net/users/index.html	Russian Fed.	GPL
3	Bodington	University of Leeds	UK	BSSL
4	Bazaar 7	University of Athabasca	Canada	-
5	CHEF	University of Michigan	USA	GPL
6	Caroline 1.4	Universit Catholique de Louvain	France	GPL
7	CassWeb 2.0	University of California Los Angeles	USA	GPL
8	Coursework	Stanford University	USA	-
9	dokeos	dokeos	Belgium	GPL
10	eClass.Net	Tulane University	USA	BSD
11	eConf	University of Namur	Belgium	GPL
12	Eduzope/Eduplone	Coalition (Infrae, Plone, others)	Europe	GPL
13	eLecture Online Lecturing System	Christian and Thomas Lang	Austria	-
14	Eedge 1.2	Chuck Wight (University of Utah)	USA	GPL
15	eTutor	University of Ottawa	Canada	GPL
16	Fle3	University of Art and Design Helsinki	Finland	GPL
17	Freestyle Learning	University of Mnster	Germany	-
18	GANESHA	Anma Formation	France	GPL
19	H2O Project	Harvard Law School	USA	GPL
20	Ilias	University of Cologne	Germany	GPL
21	Interact	Christchurch College of Education	New Zealand	GPL
22	ILE	Brad Cox	USA	BSD
23	ICR:Internet Course Reader	TeleLearning National Centre of Excellence	Canada	LGPL
24	Jones e-education V2002	Jones Advisory Group	USA	-
25	KEWL	University of Western Cape	South Africa	-
26	LearnLoop	IT university in Gothenburg	Sweden	GPL
27	LogiCampus	Tap Internet & Tarrant County College	USA	CL
28	LON-CAPA	Michigan State University	USA	GPL
29	.LRN	MIT/Heidelberg	USA/Ger	GPL
30	OLAT	University of Zurich	Switzerl and	AS
31	Manhattan Virtual Classroom 2.3	Western New England College	USA	GPL

32	MmerDesk 1.5.3.1	Ionstream	Finland	GPL
33	MnITS Internet Teaching System	Daniel Bartholomew	USA	GPL
34	Moodle 1.8	Martin Dougiamas	Australia	GPL
35	OpenCourse	Paul Jones	USA	GPL
36	O-LMS	University of Utah	USA	-
37	OpenUSS	University of Mnster	Germany	GPL
38	Segue	Middlebury College	USA	-
39	Shadownet	University of Missouri-Columbia	USA	GPL
40	Spaghettilearning.com		Italy	GPL
41	STeam	Univeristy of Paderborn	Germany	-
42	TextWeaver	San Diego State University	USA	LGPL
43	Uni Open Platform	FernUniversit?t in Hagen	Germany	GPL
44	WBT-Master	Graz University of Technology	Austria	GPL
45	Whiteboard	Todd Templeton	USA	GPL

3.5 The Future of Virtual Learning Environments

The increasing dependence on e-learning systems for learning and education through technology (today the Internet, tomorrow Digital Television) will promote further challenges to the universities in their traditional marketplace. By investing controlled resources into the future development of VLEs, institutes will be best placed to deliver learning into the next decade, as in Figure 3.4. Therefore, Web services will play an important role with this future to enhance VLE products [3, 87, 103].

One useful way of thinking about how learning environments might develop is to look at current trends in Internet technologies, and at a visual representation, as in Figure 3.4. VLEs should be modular to make future development of applications more easy and cost-effective. At the same time, the development of future teacher training should be based upon the same principals as are foreseen for future educational and training practices: general monitoring, reflection and research of emergent practices (especially of successfully incorporating technology), and individual and flexible delivery of courses and other services [3, 105].

The future of VLEs will develop high-quality pedagogical standards and method-

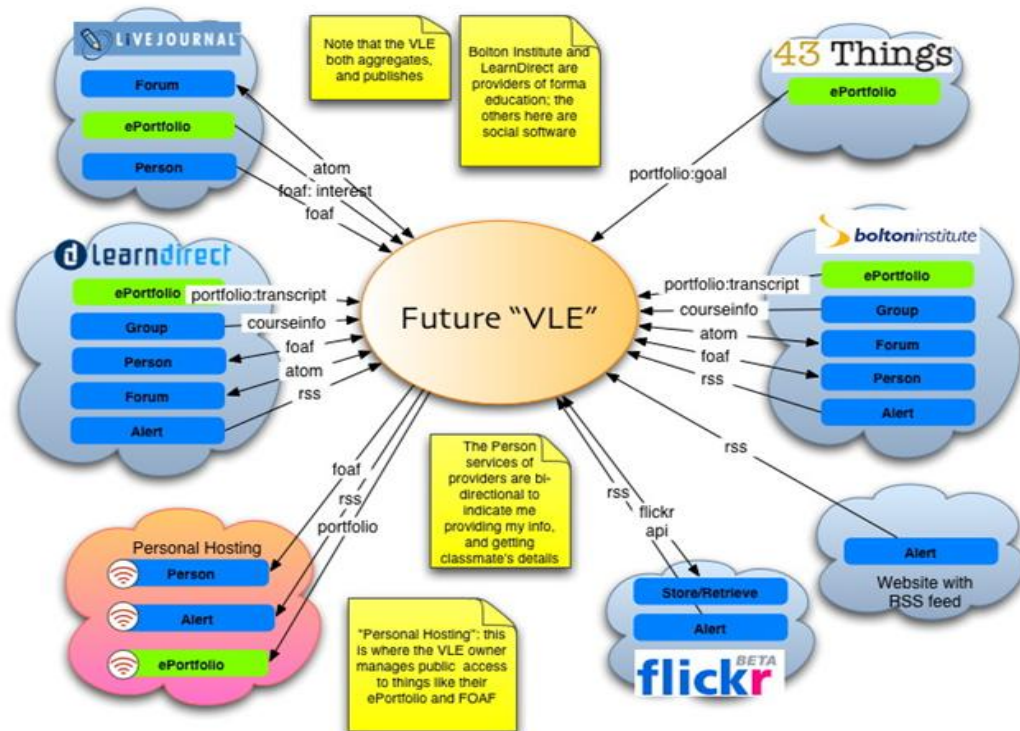


Figure 3.4: The Future of VLEs - The Visual Version [3]

ologies to support the users (teachers and students) online in relation to state-of-the-art pedagogical and didactic methods and content matters in the light of future teaching. In addition, OSS will enable the future of VLEs to make integration of generative tools, such as visualising, mind-mapping, collaborative design tools and digital libraries, all with archive capabilities. However, it is necessary to take the needs of current teachers and students into account before more extensive innovations can be incorporated [3, 106].

Currently, advances in academic computing, including the ubiquitous deployment of high-connectivity, have finally made it feasible for educators to shape the new media in transformative ways. A persistent, unified, massively multi-user and self-organizing virtual environment for learning, which capitalises on the rich collaborative capabilities of open-source technologies such as Croquet, is what is needed to take e-learning to the next plateau, enabling online ‘interactivity’ in a constructivist

sense e.g. ‘interactivity’ that is synonymous with vital, self-organizing communities of practice [96].

Dynamic OSS provides social interaction, and thus facilitates online communities. More and more online communities will come into view around the globe. Teachers should not hesitate to create and maintain their Web presence to help their learners to build computer and information literacy [2].

3.6 Summary

This chapter has described the main aspects of VLEs, especially in higher education institutes. The time is now right for educational institutions to incorporate the power of visualization technologies into the online academic area. The main reason why VLEs have become so popular in many institutions is that there are actual benefits to be gained from the use of the technology. There are many reasons why institutions are turning to OSS, but scalability and flexibility are particularly significant. Scalability enables institutions to use OSS with as many users as they like without incurring bigger license fees. Flexibility is because institutions can choose to develop the OSS to meet their particular needs. Especially in developing countries, it is now time to narrow the information gap by setting up more OSS for educators and learners.

The next chapter is important because it focuses on the system that this thesis will use and apply in its approach. This system is Moodle and this chapter gives an overview of the literature on the reasons for choosing Moodle, and the history, limitations and architecture of Moodle. The most important section in this chapter is the comparative study between Moodle and other VLE products. Also, this chapter presents the tools and activities of Moodle and concentrates on an assignments module as a specific area of study.

Chapter 4

A Comparison of Current VLE Features

Objectives

- To produce a comparative study between VLE features.
- To present the literature on Moodle as a selected platforms including its architecture, benefits, and tools.
- To focus on the reasons for choosing Moodle platform.
- To concentrate on the assignments module as a specific area of study.
- To produce websites that are using Moodle all over the World.

4.1 Introduction

Nowadays, there is an increasing demand for methodologies and technologies, especially, for e-learning. E-learning has been defined as interactive learning in which the learning content is available online and provides automatic feedback to the student's

learning activities. While recognizing that the world at large will continue to use terminology in different and often ambiguous ways, the term of Virtual Learning Environments (VLEs) is used to refer to the ‘online’ interactions of various kinds that take place between learners and tutors [107, 108, 83, 89].

There are already more than 250 providers of commercial e-learning and more than 45 of them are open-source e-learning offerings, as in the previous chapter. Perhaps the best known of these are Moodle, Ilias, Eduplone, Claroline, SAKAI, WebCT and Bscw. They have wide developer communities and present convincing arguments for considering open source as a realistic alternative to commercial products. One open-source project worthy of note, which has emerged to meet the growing interest in open-source platforms, is Modular Object-Oriented Dynamic Learning Environment (Moodle) [83, 85, 86]. Moodle is designed around pedagogical principles, namely a social constructivist philosophy using the collaborative possibilities of the Internet [109]. Moodle is free Open Source Software (OSS), which means that users are free to download, use, modify and even distribute it under the terms of GNU [107, 83, 85, 110].

This chapter is structured as follows. A comparative study between VLE features is presented in Sections 2, which is the most important section in this chapter. In Section 3, a literature review of Moodle as a selected platform is presented, including the reasons for choosing Moodle together with its limitations, the architecture of Moodle and explains most of its components. Sections 4 and 5 focus on the VLE tools of Moodle and an assignment activity respectively. Section 6 describes some of the websites that are using Moodle across the world. Finally, the summery of this chapter is in Section 7.

4.2 Comparative Study of VLE Products

An important resource for higher education, especially universities, is VLE, which has been enhancing students' progress with high quality learning around the world. This section will propose a suitable e-learning system to consider it as a specific area of study through a comparative study of the most well-known e-learning systems. It is an important to make a comparison study between VLE products to select the suitable one and test it with our approach and also explore their strengths and limitations. This comparative study is in two phases. The first phase is based on the features and capabilities of VLE tools, and the second is based on the technical aspects of the systems of VLEs.

4.2.1 Comparative Study Based on Features and Capabilities of VLE Tools

VLEs have many features and capabilities such as forums, content management, quizzes with different kinds of questions, and a number of activity modules. Moodle has an additional number of contributed modules, including SCORM WebQuest and the Document Management System [111]. In this section, we have selected 10 VLE products, including Moodle, to make comparisons between them, and our first comparison is based on the features and capabilities of VLE tools. I am very thankful to the EduTools website [112], which lists more than 80 VLE products and has performed a comparison of 42 VLE features and capabilities, as in Table 4.1.

Our comparison focuses on two kinds of products. The first is commercial e-learning systems and comprises Desire2Learn 8.1, ANGEL Learning Management Suite (7.1), TeleTOP Virtual Learning Environment, The Blackboard Learning System (V7) and Scholar360. The second is OSS and comprises LON-CAPA, Sakai 2.3, dotLRN/OpenACS, ATutor 1.5.4 and Moodle 1.8. The comparison has two an-

swers, Y or N. Y means the product has the feature and N means the product does not. Table 4.1 displays information about the ten VLE software packages used in the first comparison. VLE Tools are criteria-based products that enable developers to evaluate and select the best VLE product. No single VLE product can possibly meet all these criteria and may not be the best for interface, technical, functional, or cost reasons. These criteria are described in Table 4.2 [83].

Table 4.1: General Information about the Selected Products

No	Product	Developer name	Date	URL
1	LON-CAPA	Gerd Kortemeyer	Oct/2006	LON-CAPA Project
2	Desire2Learn 8.1	Desire2Learn Inc.	Oct/2006	Desire2Learn Inc.
3	ANGEL Learning 7.1	ANGEL Learning Inc	Oct/2006	Angel Learning
4	TeleTOP VLE	TeleTop B.V.	Oct/2006	TeleTop
5	Blackboard (V6.2)	BlackBoard	Nov/2006	Blackboard LSE
6	Sakai 2.3	Sakai 2.3	Nov/2006	Sakai
7	dotLRN/OpenACS	dotLRN	Jan/2007	dotlrn.org
8	Scholar360	Scholar360	Jan/2007	www.scholar360.com
9	ATutor 1.5.4	University of Toronto	April/2007	atutor.ca/atutor/index.php
10	Moodle 1.8	Moodlerooms	April/2007	www.Moodle.org

VLEs as an e-learning system have many features and capabilities. For simplicity, we have divided these features and capabilities into three phases, which are Learner Tools, Support Tools and Technical Tools, as in Table 4.2. Chapter 3 ‘Virtual Learning Environments As E-learning Systems’ has more details about these tools. Table 4.2 lists the features and capabilities of VLE tools that we have used in our comparison in this chapter.

4.2.1.1. Learner Tools

This phase contains three kinds of tools: Communication Tools, Productivity Tools and Student Involvement Tools. Each Learner Tool has some features and capabilities as in Table 4.2. As we can see in Table 4.3, the comparison between the VLE products is based on Learner Tools. Four products are shown to be the best

Table 4.2: Summaries of the Features and Capabilities of VLE tools

1) Learner Tools	2) Support Tools	3) Technical Specifications
1. <u>Communication Tools</u> <ul style="list-style-type: none"> • Discussion Forums • File Exchange / Internal Email • Online Journal/Notes • Real-time Chat • Video Services / Whiteboard 	1. <u>Administration Tools</u> <ul style="list-style-type: none"> • Authentication • Course Authorization • Registration Integration • Hosted Services 	1. <u>Hardware/Software</u> <ul style="list-style-type: none"> • Client Browser Required • Database Requirements • Server Software • UNIX Server • Windows Server
2. <u>Productivity Tools</u> <ul style="list-style-type: none"> • Bookmarks • Orientation / Help • Searching Within Course • Calendar / Progress Review • Work Offline/Synchronize 	2. <u>Course Delivery Tools</u> <ul style="list-style-type: none"> • Course Management • Instructor Helpdesk • Online Grading Tools • Student Tracking • Automated Testing and Scoring 	2. <u>Pricing/Licensing</u> <ul style="list-style-type: none"> • Company Profile • Costs • Open Source • Optional Extras • Software Version
3. <u>Student Involvement Tools</u> <ul style="list-style-type: none"> • Groupwork • Self-assessment • Student Community Building • Student Portfolios 	3. <u>Curriculum Design</u> <ul style="list-style-type: none"> • Accessibility Compliance • Course Templates • Curriculum Management • Customized Look and Feel • Instructional Standards Compliance • Instructional Design Tools • Content Sharing / Reuse 	

with almost the maximum number of features - 15 out of 16 features or capabilities of Learner Tools. These products are Moodle, Desire2Learn, ANGEL Learning Management Suite, and Sakai.

The LON-CAPA and dotLRN/OpenACS products have the minimum features and capabilities of Learner Tools, missing 5 out of 16 Learner Tools. TeleTOP VLE and The Blackboard Learning System have missed 2 out of 16 Learner Tools. Overall the best OSSs are Moodle, which missed 1 out of 16 Learner Tools.

Table 4.3 contains Learner Tools, which have many features and capabilities, and in order to understand what they mean, please refer to Chapter Three ‘Virtual Learning Environments as E-learning Systems’ where we explained them. We also list them in Table 4.2.

Table 4.3: The Comparison between Selected VLE Products based on Learner Tools.

No	1	2	3	4	5	6	7	8	9	10
Product Name	LON-CAPA	Desire2Learn 8.1	ANGEL Learning Management Suite	TeleTOP VLE	The Blackboard Learning System	Sakai 2.3	dotLRN/OpenACS	Scholar360	ATutor 1.5.4	Moodle 1.8
Tools	1. Learner Tools									
1.1. Communication Tools										
➤ Discussion Forums	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Discussion Management	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ File Exchange	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Internal Email	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Online Journal/Notes	N	Y	Y	N	Y	Y	N	Y	N	Y
➤ Real-time Chat	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Video Services	N	N	N	N	N	N	N	N	N	Y
➤ Whiteboard	N	Y	Y	Y	Y	Y	N	N	Y	Y
1.2. Productivity Tools										
➤ Bookmarks	Y	Y	Y	Y	N	Y	Y	Y	N	N
➤ Calendar / Progress review	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Orientation/Help	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
➤ Searching Within Course	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
➤ Work Offline/Synchronize	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
1.3. Student Involvement Tools										
➤ Groupwork	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
➤ Student Community Building	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Student Portfolios	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
Total Features	16	16	16	16	16	16	16	16	16	16
Total Available Features	11	15	15	14	14	15	11	12	12	15
Total Missing Features	5	1	1	2	2	1	5	4	4	1

4.2.1.2 Support Tools

These tools contain three kinds of tools: Administration Tools, Course Delivery Tools, and Content Development Tools, and all of these tools have features and capabilities.

As we can see in Table 4.4, this comparison between the VLE products is based on Support Tools. In this phase, all products have all features and capabilities except Scholar360, TeleTOP Virtual Learning Environment and The Blackboard Learning System (V.7). This means that Moodle and the other remaining products are strong on Support Tools.

The Learner Tools in Table 4.4 have many features and capabilities, and to understand what they mean, please refer to Chapter Three ‘Virtual Learning Environments as E-learning Systems’. We have also listed them in Table 4.2.

Table 4.4: The Comparison between Selected VLE Products based on Support Tools.

No	1	2	3	4	5	6	7	8	9	10
Product Name	LON-CAPA	Desire2Learn 8.1	ANGEL Learning Management Suite	TeleTOP VLE	The Blackboard Learning System	Sakai 2.3	dotLRN/OpenACS	Scholar360	ATutor 1.5.4	Moodle 1.8
Tools										
2. Support Tools										
2.1. Administration Tools										
➤ Authentication	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Course Authorization	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Hosted Services	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Registration Integration	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
2.2. Course Delivery Tools										
➤ Test Types	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Automated Testing Management	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Automated Testing Support	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Course Management	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Online Grading Tools	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Student Tracking	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
2.3. Content Development Tools										
➤ Accessibility Compliance	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
➤ Content Sharing/Reuse	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
➤ Course Templates	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Customized Look and Feel	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Instructional Design	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
➤ Instructional Standards Compliance	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Total Features	16	16	16	16	16	16	16	16	16	16
Total Available Features	16	16	16	15	15	16	16	15	16	16
Total Missing Features	0	0	0	1	1	0	0	1	0	0

4.2.1.3 Technical Specifications Tools

These tools contain two kinds of tools: Hardware/Software Tools and Pricing/Licensing; all kinds of Technical Specifications Tools have some features and capabilities, as in Table 4.5. The costs feature is different from other features because if the product has no cost, it means that product has an advantage and we will calculate it as Yes (Y). For example, in Table 4.5, Moodle has two N and we calculated N of cost as Y, so in the final score Moodle has missed just one feature.

As we can see in Table 4.5, the comparison between the VLE products is based on Technical Specifications Tools. In this phase, the best product is ATutor 1.5.4, Moodle 1.8, Scholar360 and The Blackboard Learning System, which missed only 1 out of the 8 Technical Specifications Tools. The weakest product is LON-CAPA, which missed 5 out of the 8.

Table 4.5: The Comparison between Selected VLE Products based on Technical Specifications Tools.

No	1	2	3	4	5	6	7	8	9	10
Product Name	LON-CAPA	Desire2Learn 8.1	ANGEL Learning Management Suite	TeleTOP VLE	The Blackboard Learning System	Sakai 2.3	dotLRN/OpenACS	Scholar360	ATutor 1.5.4	Moodle 1.8
Tools										
3. Technical Specifications										
3.1. Hardware/Software Tools										
➤ Client Browser Required	N	Y	Y	N	Y	Y	N	Y	Y	Y
➤ Database Requirements	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
➤ Unix Server	Y	N	N	Y	Y	Y	Y	Y	Y	Y
➤ Windows Server	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
3.2. Pricing/Licensing Tools										
➤ Company Profile	N	Y	Y	N	Y	Y	N	Y	Y	N
➤ Costs	N	Y	Y	Y	Y	N	N	Y	N	Y
➤ Open Source	Y	N	N	N	N	Y	Y	N	Y	Y
➤ Optional Extras	N	Y	Y	Y	Y	N	N	Y	Y	Y
Total Features	8	8	8	8	8	8	8	8	8	8
Total Available Features	3	6	6	4	7	6	4	7	8	7
Total Missing Features	5	2	2	4	1	2	4	1	1	1

The Technical Specifications in Table 4.5 have many features and capabilities, and to understand what these features and capabilities mean, please refer to Chapter Three ‘Virtual Learning Environments as E-learning Systems’. We have also listed them in Table 4.2.

4.2.1.4 The Final Result of the Comparison between the Ten VLE Products

From Table 4.6, we can see the final result of the comparison between the ten VLE products. The best product is Moodle 1.8, which has missed just 2 out of 40 features and capabilities, and the second products are Desire2Learn 8.1, ANGEL Learning Management Suite (7.1) and Sakai 2.3 equally, which have missed 3 out of the 40. Also, Moodle is the best of the OSS products. The weakest product is LON-CAPA, which has missed 10 out of the 40.

We use the GraphPad Prism software to analyse, graph and present scientific

Table 4.6: The Final Result of the Comparison between Ten VLE Products.

No	1	2	3	4	5	6	7	8	9	10
Product Name	LON-CAPA	Desire2Learn 8.1	ANGEL Learning Management Suite (7.1)	TeleTOP Virtual Learning Environment	The Blackboard Learning System (V 7)	Sakai 2.3	dotLRN/OpenACS	Scholar360	ATutor 1.5.4	Moodle 1.8
Tools										
Total Features	40	40	40	40	40	40	40	40	40	40
Total Available Features	30	37	37	33	36	37	31	34	35	38
Total Missing Features	10	3	3	7	4	3	9	6	5	2

data of VLE products because it is a powerful combination of basic biostatistics, curve fitting and scientific graphing in one comprehensive program. It has been dedicated to creating software exclusively for the international scientific community. More than one hundred scientists in over one hundred countries rely on Prism to analyse, graph and present their scientific data. Since 1984, created by scientists for scientists, Prism's intuitive programs have provided researchers worldwide with the tools they need to simplify data analysis, statistics and graphing [113].

Figure 4.1 shows the comparison between the ten products of VLE systems. The total features are 40 but no product has reached this number. In Figure 4.1, P1, P2 etc mean the VLE product as mentioned in Table 4.6 respectively.

As in Figure 4.1, the best VLE product is P10 (Moodle 1.8), which has 38 out of 40 features and capabilities, and the weakest is P1 (LON-CAPA), which has 30 out of the 40. P10 (Moodle) has 38 out of the 40 features and capabilities and is the number 1 out of the 10 VLE products. It is number 1 out of the OSS products, which itself has missed just 2 out of the 40 features and capabilities.

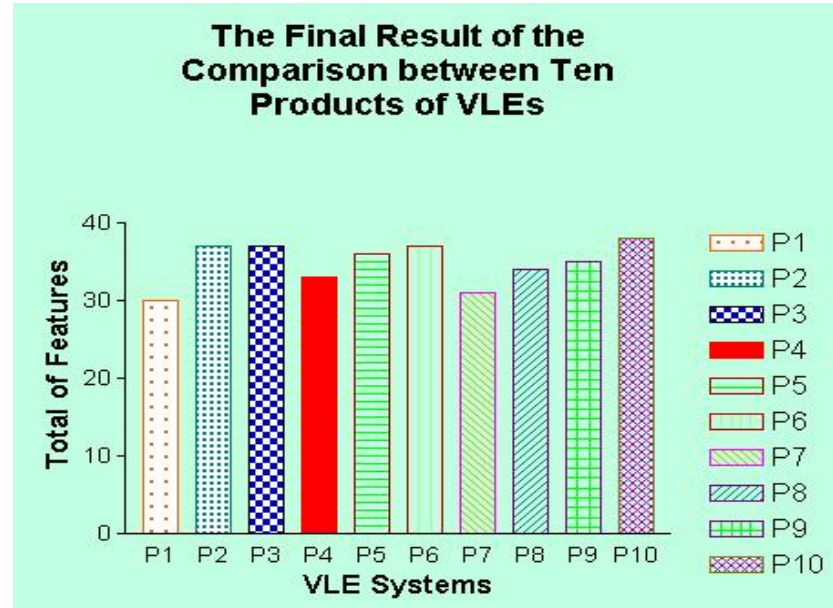


Figure 4.1: The Total Features of the Ten VLE Products

4.2.2 Comparison Based on Focusing on the Technical Aspects of the VLE Systems

In this session, the comparison between the systems is based on technical categories. All VLE systems will be compared with the Moodle system as part of our study. As in our literature review, we have selected four studies focusing on this kind of comparison.

4.2.2.1 First Study

As in [114], Moodle has limitations, notably it lacks SCORM support, and its roles and permissions system is limited. However, these limitations can be fixed, and are part of the project roadmap in Moodle site.

Table 4.7 reveals that ATutor, while strong in features and usability, has serious architectural limitations, and although some features in ATutor warrant further investigation, it may be that candidates will opt for Moodle.

ILIAS, while promising, has a complex architecture with tight coupling that is

Table 4.7: Comparison based on focusing on the technical aspects of the VLE systems

Category \ Product	ATutor	ILIAS	Moodle
Architecture	Weak	Complex	Good
Implementation	Weak	Complex	Good
Interoperability	Bad	Good	Average
Cost of ownership	Medium	High	Low
Strength of the community	Low	Medium	High
Licensing	GPL	GPL	GPL
Internationalization	Weak	Average	Good
Accessibility	Excellent	Bad	Average
Document transformation	No	Average	No

hard to work with and debug. The code is new, and lacks maturity. The developer community of ILIAS is small outside the core team. Nevertheless, some features in ILIAS deserve to be reviewed before opting for Moodle.

Moodle has a good architecture, implementation, inter-operability, and internationalization, and also has the strength of the community. It is free and its accessibility is average. On the other hand, it has limitations, as mentioned above.

4.2.2.2 Second Study

Table 4.8 shows the comparison between 4 products of VLE systems. The comparison is based on categories as [115] determined. This study has proved that Moodle outperforms all other systems and scored 4.467 out of 5. In contrast, Boddington gained the lowest score, at 2.439.

Moodle has nearly the maximum score because it has many of the features expected from an e-learning platform, including forums, resources, quizzes with different kinds of questions, and a number of activity modules. Furthermore, Moodle is very beneficial for language teaching and learning because the interactive tools, such as wiki, discussion forums, and quizzes, can be selectively employed to meet the objectives of the course and to motivate students.

Table 4.8: Comparison based on focusing on some Features and Categories of VLEs.

Category \ Product	Moodle	Sakai	ATutor	Boddington
Functionality	1.25	.75	.25	.25
Usability	.8	.8	.6	.65
Documentation	.645	.465	.54	.54
Community	.6	.384	.24	.288
Security	.42	.34	.28	.42
Support	.4	.15	.35	.15
Adoption	.352	.336	.208	.336
Total Score (out of 5)	4.467	3.225	2.468	2.439

4.2.2.3 Third Study

In [97], the study reports that the result of the evaluation shows that Moodle has the best rating in the adaptation category; it can be seen in Table 4.11 as the best system concerning adaptation issues. It dominates the evaluation by achieving the best value five times. The strengths of Moodle are the realization of communication tools, the creation and administration of learning objects, the comprehensive didactical concepts and the tracking of data. In addition, the outstanding usability of Moodle leads to the maximum evaluation value in the usability category. Concerning the other platforms, ILIAS obtained the best values in the categories for technical aspects, administration, and course management.

Table 4.9: Results of the Adaptation Category

Feature \ Product	Adaptability	Personalization	Extensibility	Adaptively	Ranking
1 ATutor		#	#		3
2 Dokeos		0	*	+	2
3 dotLRN	+	+	*	0	2
4 ILIAS	+	#	*	0	2
5 LON-CAPA	+	#	#		2
6 Moodle	#	+	*		1
7 OpenUSS	#	#	#	0	2
8 Sakai	0	0	*	0	3
9 Spaghettilearning	+	#	+	0	3

Moodle has gained the best results, especially in the specific adaptation evaluation as in Table 4.9. It supports an adaptive feature called ‘lesson’ where learners can be routed automatically through pages depending on the answer to a question after each page. Furthermore, the extensibility is supported very well by a documented API, detailed guidelines, and templates for programming. In addition personalization and adaptability features are present in Moodle [97].

4.2.2.4 Fourth Study

In [116], the study reports the percentage of universities that are developing or using e-learning frameworks. The result of e-learning survey conducted the use of web-based learning management systems for higher education. This study discussed with some members of the scientific community on this field in the Department of Computer Science at the University of Oviedo.

As we can see in Figure 4.2, Moodle is the best product that has 34.55% and the second product is WebCT/Blackboard that has 27.27%. The weakest product is Sakai and dotLRN that have same percentage 10.91%.

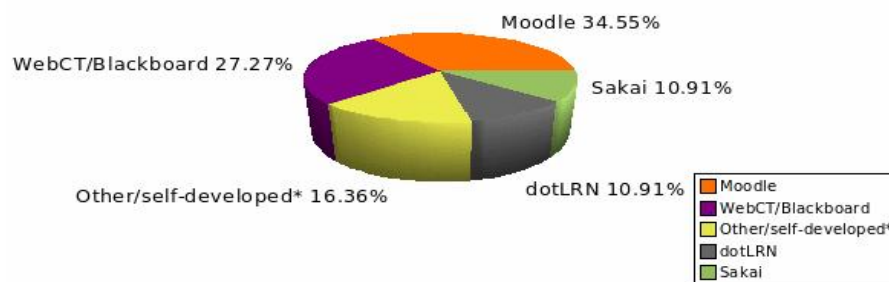


Figure 4.2: Chart of use of web-based e-learning systems in Universities

4.3 Moodle as a Selected Platform

According to the comparison study above, we have chosen Moodle as the suitable platform for this project. In this section, we will present the literature on Moodle including its architecture, benefits and limitations and tools, as well as we will mention more reasons for choosing this platform. Moodle is the most user-friendly and flexible free open-source courseware products available all over the world. Moodle is a VLE that lets teachers provide and share documents, assignments, quizzes, forums, chats, etc. with students in an easy-to-learn and user-friendly interface. Moodle is CMS designed to help educators who want to create quality online courses [83, 114, 117]. It has excellent documentation, strong support for security and administration, and is evolving towards Information Management System/Shareable Content Object Reference Model (IMS/SCORM) standards [109]. Moodle has a strong development and large user community and users can download and use it on any computer they have at hand [85, 110].

4.3.1 The Definition of Moodle

An important feature of the Moodle is the Moodle.org web site, which provides a central point for information, discussion and collaboration among Moodle users, who include system administrators, teachers, researchers, instructional designers and of course, developers. Like Moodle, this site is always evolving to suit the needs of the community. Moodle is now used not only in universities, but also in high schools, primary schools, non-profit organizations, private companies, and by independent teachers and even home-schooling parents. A growing number of people from around the world are contributing to Moodle in different ways [107, 83, 118].

Moodle is based on Social Constructionist Pedagogy, which is a learner-oriented philosophy and most VLE modules are based on it. They are largely concerned with

how course contents are delivered, in which students are involved in constructing their own knowledge [117, 97]. The learner-oriented philosophy of learning is that learners actively construct new knowledge by tinkering and experimenting, and they learn even more by explaining what they have learned to others and by adopting a more subjective stance to the knowledge being created. These ideas run parallel to the way open-source development works, in which the developers also are often users, everyone is free to tinker with the software and code is re-constructed or adapted, peer-reviewed and refined by the means of open discussion [85, 79].

The attraction of Moodle is that can use on almost servers that can use PHP. It is OSS, which means users are free to download, use, modify and even distribute it under the terms of GNU. Moodle runs without modification on Unix, Linux, Windows, Mac OS X, Netware and any other system that supports PHP, including most web host providers. Data can be stored in a single database (MySQL and PostgreSQL), but it can also be used with Oracle, Access, Interbase, ODBC and others [107, 109, 119, 79].

4.3.2 The History of Moodle

Moodle was created by Martin Dougiamas while working on a postgraduate degree at the Curtin University of Technology in Australia. In 2002, he was a Webmaster of a university and a system administrator of WebCT installation. He started to develop Moodle to solve some problems with WebCT. The original version was targeted for small classes and a case study, but steadily many features were added by developers and other contributors from all over the world [118, 111, 120].

Abhijeet Chavan and Shireen Pavri have said, ‘Moodle was born out of a need to scratch an itch. Frustrated by proprietary alternatives, Martin, then a PhD candidate in Education with a background in computer science, started Moodle in 1999. In 2002, Version 1.0 was released. Since then, Moodle has continued to

evolve at a rapid rate, managed by Martin and propelled by an active world-wide community of users and developers' [79].

The Moodle Company 'Moodle.com' has been providing managed hosting and consulting services since 2003. Currently, Moodle has a large and diverse user community with over 550,000 users on this site, speaking 75 languages in 193 countries around the world [83].

4.3.3 The Reasons for Choosing Moodle

OSS is rapidly developing, and new alternatives for non-profit organizations are emerging and maturing. Additionally, open-source learning environments such as Moodle are becoming widely adopted by university and educational institutions. Managing an LMS can be a complex task. Moodle does not hide this complexity and its detailed on-line help, examples and sensible defaults assist users in installing, administering and using the LMS. Moodle allows users to post news items, assignments, electronic journals and resources, and to collect assignments etc. The greatest strength of Moodle is the community that has grown around the project. Both developers and users participate in Moodle's active discussion forums, sharing tips, posting code snippets, helping new users, sharing resources and debating new ideas [83, 109, 117, 119, 118, 79, 121].

Thus, we have chosen the Moodle software to be the area of study and analysis. We want to understand Moodle's environment to explore its functionalities and limitations in order to develop practical examples of the use of VLEs over the world. We list here the most important reasons for choosing this package:

1. Moodle is OSS, which means users are free to download it, use it, modify it and even distribute it under the terms of GNU [107, 83, 119, 79, 109, 122];
2. Moodle is CMS & VLE, and lets teachers provide and share documents, graded

assignments, quizzes, discussion forums, etc. with their students in an easy-to-learn manner and to create quality online courses [83, 85, 120];

3. Moodle can be used on almost all servers that can use PHP [79, 121];
4. The key to Moodle is that has been developed with both pedagogy and technology in mind. One of the main advantages of Moodle over other systems is a strong grounding in social constructionist pedagogy with good educational tools [119, 123, 2];
5. It works well with languages and is currently being used in 75 languages in 193 countries [83];
6. It has excellent documentation, and strong support for security and administration and easy to upgrade from one version to the next[83, 85];
7. It has many user-friendly features such as easy installation, customization of options and settings, good support/help and good educational tools [2].
8. It demonstrates the use of OSS in creating a high quality e-learning environment that incorporates many other subjects [122, 114];
9. Moodle is the LMS most often recommended of all the OSS, as well as being the most popular. Therefore, the credibility of Moodle is very high. At present, there are 52289 web sites from 193 countries that have registered with it [79];
10. The importance of Moodle is its good reputation according to good reports, grade of admission and number of places, existing languages, etc;
11. Moodle should be able to be used in conjunction with other systems. It keeps all files for one course within a single, normal directory on the server. Administrators allow the provision of seamless forms of file-level access for each teacher, such as SMB, FTP, and so on. Currently, there is work on more

features planned for Moodle in future versions, such as export and import data using XML-based formats that can be integrated visually into other web sites. In addition, this thesis has presented a good solution for this integration, enabling more VLEs to work together by using Web services technology;

12. Moodle runs without modification on Unix, Linux, Windows, Mac OS X, Netware and any other systems that support PHP. Data is stored in a single database: MySQL or PostgreSQL are best but it also supports Oracle, Access, Interbase, ODBC and others [122];
13. Moodle is the core LMS for the New Zealand Open Source Virtual Learning Environment project[114].
14. British Columbia Distance Learning Program [124] has recommended Moodle and said:

‘We recommend that BC campus support a provincial implementation of Moodle for all public educational institutions at the same level of support as currently provided for WebCT. Based on the investigations and implementations of Moodle at the local level of institutions involved in this collaborative project, as well as a province-wide Moodle pilot, we believe that Moodle is a viable alternative to WebCT as an on-line Course Management System (CMS)’ [124, pp.1].

15. Some universities integrate Moodle with other VLE products, such as Oxford University which has integrated two OSS learning environments, Bodington VLE and Moodle [125], although they are slightly different to each other;
16. Itmazi [121], who is working at the Open University UK, has said:

‘According to the past review (of 58 studies of comparison and evaluation, Moodle is the LMS most recommended of all the packages

of OSS, besides, Moodle is the most popular LMS of all the bundles of OSS (it depends on the number of competitions times)' [121, pp.144].

4.3.4 The Limitations of Moodle

Moodle's low cost, flexibility and ease of use helps bring LMS technology within the reach of those with limited technical and financial resources. Moodle is a fine example of how and why open source works [119]. On the other hand, Moodle has some disadvantages and we will mention some of them, as follows:

1. OSS is only for IT experts and is too difficult for normal users to install and use; more than 66% users of Moodle have identified themselves as teachers, on-line learning researchers or educational administrators [79, 120];
2. Lack of simple-to-obtain support. The forum has a great deal of information, but nearly all forums are in the English language [79];
3. It requires that someone on staff takes responsibility for making it work, you cannot just telephone Moodle technical support;
4. Although good with languages, some developments may be needed for vigorous handling of MathML and enhanced tracking features. Still, this program receives a high recommendation [114, 120];
5. The Moodle website states that the steps required for getting Moodle up and running on a web server are very simple, but in practise this is not the case. There have been many problems that we have had to overcome, which required a technical understanding of the underlying technology and the way it all hangs together [122].

4.3.5 Architecture of Moodle Sites

The strength of Moodle is its simple, but solid design and architecture developed by Martin Dougiamas. The architecture of Moodle sets an excellent foundation, following good practices of low coupling and high cohesion, which the other LMSs fail to achieve. This yields a system that is simple, flexible and effective and easily accessible to developers [83]. Figure 4.3 illustrates the architecture of the Moodle site and its components.

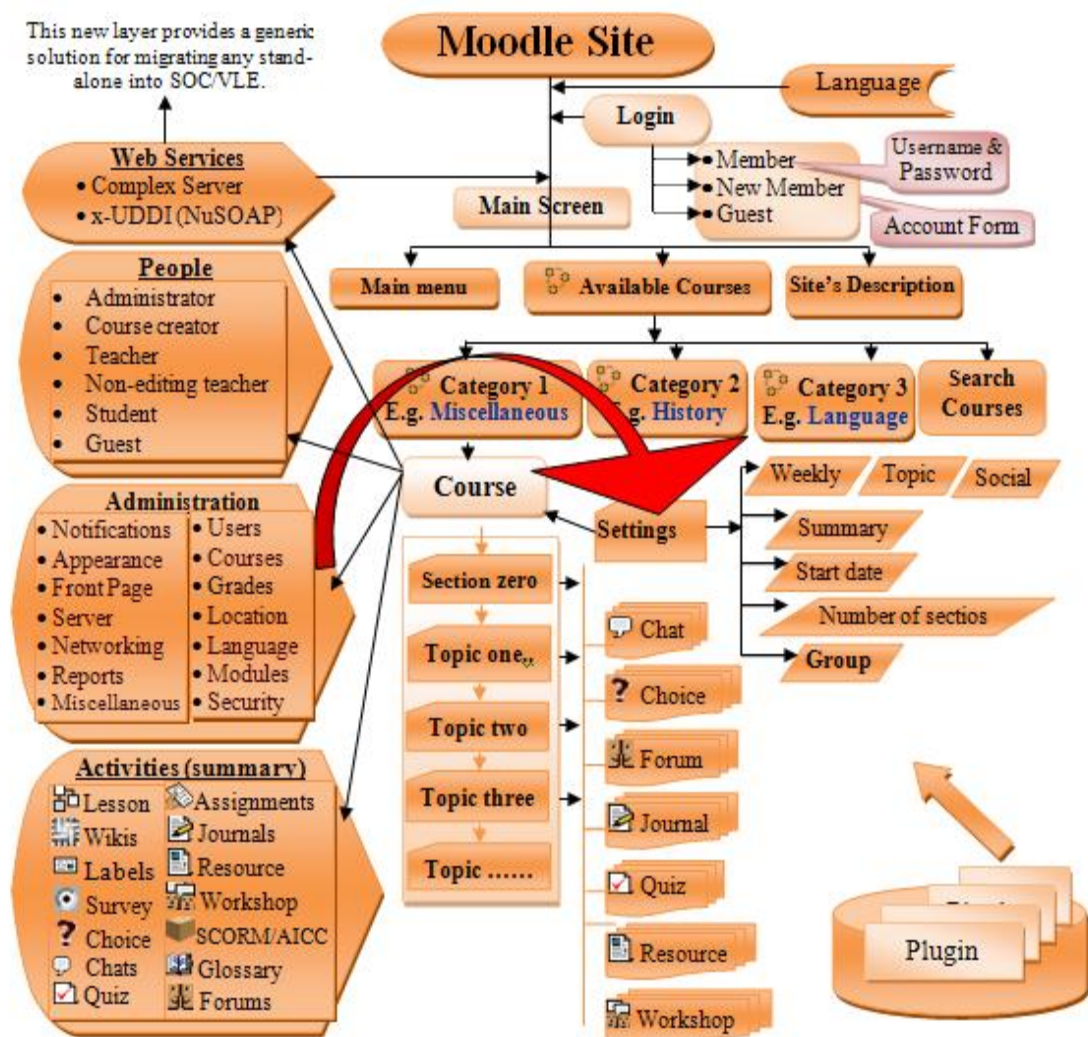


Figure 4.3: Architecture of Moodle Sites

Moodle is a huge VLE software and it is difficult to focus on all parts in this

chapter. Therefore, we will focus on the most important components, as in Figure 4.3, and these components are:

4.3.5.1. Main Screen

This screen is the main page of the Moodle site, and the administrator can control this page by the Front Page button as in figure 4.4. This screen has choices that the administrator can offer Moodle's users; options such as site administration, categories, courses, calendar and upcoming events, as well as the option to write an introduction.



Figure 4.4: The Main Screen

4.3.5.2. People

The different kinds of users in any VLE platform and especially in Moodle are described in Table 4.10. These users need especial authorization depending on their

level in Moodle. For example, an administrator has the full permission to do anything in the Moodle site and can control users and courses.

Table 4.10: The Users in Moodle Software

No	User	Description
1	Administrator	This kind of user is the most important user and has the full permission to do anything in Moodle, especially in courses. It has the responsibility to manage the site and control all users.
2	Course creator	This user can create new courses and manage them as well as teach these courses.
3	Teacher	Can do anything within a course, including changing the activities and grading students.
4	Non-editing teacher	Can teach in courses and grade students, but may not alter activities.
5	Student	Students generally have fewer privileges within a course.
6	Guest	Has minimal privileges and usually cannot enter text anywhere.
7	Authenticated user	All logged in users.

The authorization for users in Moodle is divided into six levels, as in Figure 5.9. Every user has some level of permission that Moodle permits, and Figure 5.9 displays this permission in percentage terms. Administrator has 100% and can do anything in the site. In contrast, Guest has just 10%, such as looking at available courses and sharing in general forums.

4.3.5.3. Activities

Moodle contains a wide range of activity modules, which are activity modules, resource types, and open source and they can be used to build up any type of course. In activity modules, there is assignment module, which will be the area of focus for our study. The study is aimed at analysing and evaluating the current Moodle, focusing on assignment activity to explore its functionalities and limitations as in Section 9 in this chapter. All these activities are under course in the Moodle architecture, as in Figure 4.3. These activities are described in Section 8 of this Chapter in more details.















4.3.5.4. Web services

Web services are a new technology and they have made important contributions to knowledge, especially to the e-business field. Therefore, it is important to use this technology in non-profit organizations such as e-learning. This thesis has succeeded in using Web services together with VLE platforms, by using Moodle as a case study. Chapters 5 and 8 have more information on how to use this technology with Moodle.

4.3.5.5. Administration

Administration has many tools related to course, as in Table 4.11. An administrator can control the course through the administration tools but can only control these tools by the permission granted. In our example, all these tools appear for the administrator in the administration site, as in Table 4.11.

Table 4.11: The Tools in Administration Form in Course in Moodle Sites

No	Tool	Description
1	 Turn editing on	This tool enables the user to open and activate a course.
2	 Settings	To create a new course.
3	 Assign roles	To add users for the level of authorisation as in Figure 4.2.
4	 Groups	The group mode can be defined at two levels: course level and activity level, and each can be sub-divided into three: no groups, separate groups and visible groups.
5	 Backup	To save and backup data in user's computer
6	 Restore	This allows users to restore files to a course.
7	 Import	To upload files that have been saved in a computer with the export feature.
8	 Reset	This tool allows users to empty a course of user data, while retaining the activities and other settings.
9	 Reports	These are available for each participant and they show their activities in the current course. Teachers always have access to these reports, using the button visible on each person's profile page. Student access to their own reports is controlled by the teacher via a course setting
10	 Questions	Questions five types multiple choice, short answer, true/false, matching and numerical. These questions have an option that is activated by clicking on the checkbox.
11	 Scales	Teachers can create new custom scales to be used in a course for any grading activities.
12	 Files	To upload files
13	 Grades	Many of the activities allow grades to be set. By default, the results of all grades within the course can be seen in the Grades page, available from the main course page.
14	 Unenrol me from 1101	This means take a user out of course.

4.4 Virtual Learning Environment Tools of Moodle

This section presents the main tools of Moodle, which are activity modules, resource types, and open source. In activity modules, there is assignment module, which will be the area of focus for our study. The study is aimed at analysing and evaluating the current Moodle, focusing on assignment activity to explore its functionalities and limitations. In Section 4.9 in this chapter, we will discuss assignment activity in more detail. All these activities are under course in the Moodle architecture, as in Figure 4.3. These activities are:

4.4.1 Activity Modules

Moodle contains a wide range of activity modules that can be used to build up any type of course. These activities provide a central point for information, discussion and collaboration among Moodle users. The current activates as in version 1.8 are as follows [83]:

Assignments: The assignment module enables a teacher, in a simple way, to allow students to upload and prepare any digital content for grading. In Section 4.9, we will focus on assignment activity and go more deeply to discover all its features.

Chats: These allow participants to have a real-time synchronous discussion via the web. This is a useful way to get a different understanding of each other and the topic being discussed.

Choices: These enable teachers to ask questions, and they specify a choice of multiple responses.

Forums: These are where the most discussion takes place between users. Here are some of the advantages and disadvantages of forums as in Table 4.12:

Table 4.12: Advantages and Disadvantages of Forums in VLEs

Advantages	Disadvantages
❖ Encourages peer-to-peer support.	❖ Need time to learn new teaching skills.
❖ Pupils comfortable with informal communication styles.	❖ Some pupils see forums as playgrounds.
❖ Asynchronous learning is time efficient.	❖ Forums need admin and moderation.
❖ Synchronous use offers different options.	❖ Forums can be bandwidth hungry and potentially expensive to run.
❖ Pupil responsibility opportunities.	❖ Online security is a concern.
❖ Builds a body of shared knowledge.	

Glossary: This allows participants to create and maintain a list of definitions, like a dictionary, and enables teachers to export entries from one glossary to another within the same course.

Journal: This is private between student and teacher and each journal can be directed by an open question. For each particular journal, the whole class can be assessed on one page in one form. In journal, it easy to publish ideas, can set up group blogs, pupils can help each other [122].

Lesson: This delivers content in an interesting and flexible way. It consists of a number of pages; each page normally ends with a question and a number of possible answers.

Quizzes: This module allows the teacher to design and set quiz tests, consisting of multiple choices, true/false and short answer questions. Quizzes can allow multiple attempts and they are automatically marked, and the teacher can choose whether to give feedback or to show correct answers.

Resources: These contain information that the teacher wants to bring into the course.

Wikis: This enables documents to be authored collectively in a simple markup language using a Web browser. ‘Wiki wiki’ means ‘super fast’ in the Hawaiian language, and it is the speed of creating and updating pages that is one of the defining aspects of wiki technology. The Moodle Wiki module enables participants

to work together on web pages to add, expand and change the content.

4.4.2 Resource Types

Moodle, as any VLE system, supports a range of different resource types that allow users to insert almost any kind of web content into courses, and these resources are under course [83].

Text Page: This is a simple page written using plain text. A number of formatting types are available to help turn plain text into attractive web pages.

HTML Page: It is easy to develop a complete single web page within Moodle, especially when users are using Moodle's WYSIWYG HTML editor.

Files and Web Pages: These allow users to link any web page or other file on the public Internet as well as any web page or other file that users have uploaded into the course files area from their own desktop computer.




Directory: This can display a whole directory (and its subdirectories) from the course files area. Students can then browse and view all of those files.












4.5 Assignments Module















Firstly, some general information about the Assignment Module will help understand how to apply its concepts. It gives teachers an easy-to-use method for their students to upload digital content for grading or for storing on their hard drives, such data can be submitted quickly in response to an assignment such as essay, audio, video, etc. It does not necessarily require uploads. Teachers can create offline assignments to remind their students of real-world assignments that they need to complete. Assignment can be used to record grades online for activities that do not have an online component[83, 126]. The main screen of assignment is as in Figure 4.5:

Assignment name:


Description:

[Write carefully](#)  [Ask good questions](#)  [About the HTML editor](#) 

Trebuchet MS 3 (12pt) Normal **B** *I* U           


Path: body


Grade: 

Available from: ☒ 1 July 2007 - 04 55

Due date: ☒ 8 July 2007 - 04 55

Prevent late submissions: No

Assignment type: Upload a single file 

Group mode: No groups 

Visible: Show

Figure 4.5: The Main Screen of Assignment Module in Moodle

4.5.1 Assignment Types

Assignment type allows teachers to grade various types of student submissions. There are four standard assignment types as in Figure 4.6:

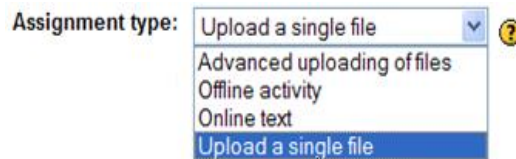


Figure 4.6: Assignment Types in Moodle

4.5.1.1 Offline Activity Assignment

This is useful when the assignment is performed outside Moodle; it could be something elsewhere on the web or face-to-face. Students can see a description of the assignment but cannot upload files. Grading works normally, and students receive notifications of their grades [83].

4.5.1.2 Online Text Assignment

Teacher/Student can edit a text using the normal editing tools. Teachers can grade them online, and even add in-line comments or changes. This kind of assignment gives the user more options, as in Figure 4.7.

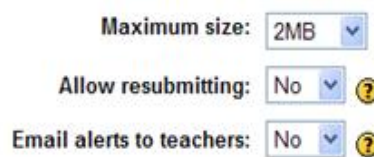
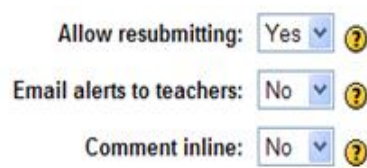


Figure 4.7: Online Text Assignment Type in Moodle

4.5.1.3 Upload Single File Assignment

This type of assignment allows each participant to upload a single file, of any type. This might be a Word processor document, or an image, a zipped web site, or anything they have been asked to submit. Also, this kind of assignment gives the user more options as in online text assignment, as in Figure 4.8.



Allow resubmitting: Yes ▾ ?

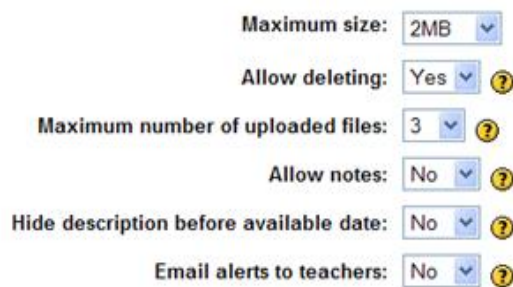
Email alerts to teachers: No ▾ ?

Comment inline: No ▾ ?

Figure 4.8: Upload Single File Assignment Type in Moodle

4.5.1.4 Advanced uploading of files

This type of assignment allows each participant to upload one or more files in any format such as Word documents, images, web sites, and zipped files and so on. Participants may also enter notes describing the submitted files, progress status or any other text information. Submission of this type of assignment must be manually finalized by the participant. Users can review the current status at any time; unfinished assignments are marked as Draft. Users can revert any ungraded assignment back to draft status. Also, this kind of assignment gives the user more options as in online text assignment, as in Figure 4.9[83].



Maximum size: 2MB ▾

Allow deleting: Yes ▾ ?

Maximum number of uploaded files: 3 ▾ ?

Allow notes: No ▾ ?

Hide description before available date: No ▾ ?

Email alerts to teachers: No ▾ ?

Figure 4.9: Advanced Uploading of File Assignment Type in Moodle

Assignment activity has one main class containing four main standard assignment classes (in version 1.8). Assignment is an activity under course and this means that users cannot access this activity without accessing the course, as in Figure 4.10. A course has many activities (as in Section 4.8.1 earlier in this chapter) and the assignment is one of them.

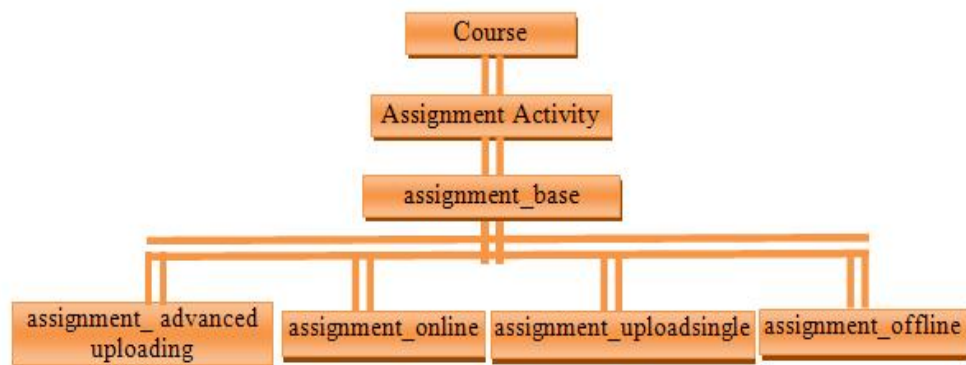




















































Figure 4.10: The Architecture of Assignment Module in Moodle

The above diagram illustrates all the main classes within the assignment model. The main class (`assignment_base`) has 55 functions. The sub-classes (`assignment_uploadsingl`, `assignment_online` and `assignment_offline`) have 5, 3 and 8 respectively. Also, there are two other sub-classes (`backuplib` and `restorelib`) that have 11 and 7 functions respectively.

4.6 Websites are using Moodle all over the World

Moodle has a large and diverse user community with over half a million registered users on this site alone, speaking 75 languages, and currently 41906 Moodle sites from approximately 200 countries have been registered [83, 110]. Table 4.13 shows 50 countries, chosen from those 200 countries, that have sites and have registered with Moodle.

Table 4.13: The number of websites using Moodle in some countries around the world.

No	Country	No of Sites	No	Country	No of Sites
1	United States 	10979	26	Russian Federation 	374
2	Spain 	3905	27	Indonesia 	327
3	United Kingdom 	3876	28	Venezuela 	304
4	Brazil 	2718	29	New Zealand 	282
5	Germany 	2480	30	Sweden 	263
6	Portugal 	1923	31	Turkey 	253
7	Australia 	1396	32	Malaysia 	246
8	Canada 	1343	33	Hungary 	234
9	Mexico 	1306	34	Belgium 	200
10	Italy 	1219	35	Greece 	190
11	Poland 	1020	36	South Africa 	177
12	Colombia 	943	37	Korea 	146
13	Thailand 	914	38	Ukraine 	134
14	Taiwan 	834	39	Hong Kong 	130
15	France 	795	40	Romania 	120
16	Austria 	736	41	Denmark 	113
17	Netherlands 	714	42	United Arab Emirates 	99
18	Chile 	706	43	Egypt 	98
19	Japan 	692	44	Iran 	84
20	Peru 	657	45	Bulgaria 	87
21	Argentina 	651	46	Saudi Arabia 	86
22	Switzerland 	542	47	Morocco 	54
23	Finland 	440	48	Lebanon 	29
24	China 	414	49	Syrian 	7
25	India 	384	50	Yemen 	6

The highest number of sites using Moodle is in the United States of America, where there are more than 10979 sites. In Spain and the United Kingdom, there are more than 3905 and 3876 sites respectively that have installed and are using Moodle. In contrast, in Yemen there are only 6 sites using Moodle. Figure 4.11 illustrates Moodle on a map of the world. We can see from this map that Moodle is concentrated in Europe and the United States.

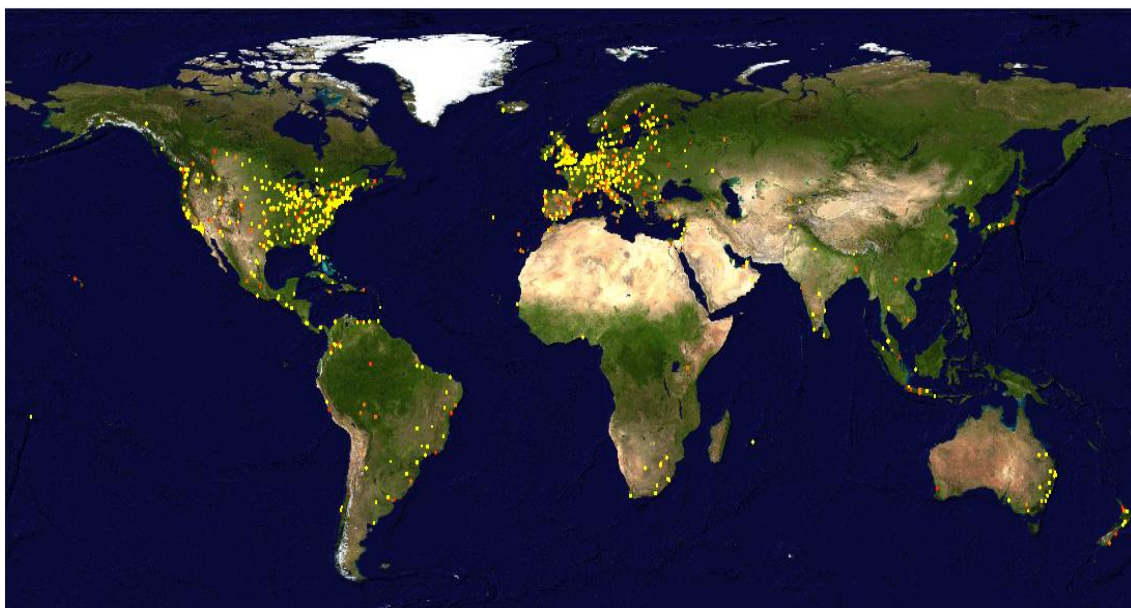


Figure 4.11: Moodle Sites that have registered with Moodle across the world.

4.6.1 Moodle in the Kingdom of Saudi Arabia

Education in Saudi Arabia has been increasing over recent years, and therefore an increasing number of institutions have been adopting e-learning systems. Figure 4.12 demonstrates the increasing use of the Moodle system in Saudi Arabia over the last six years.

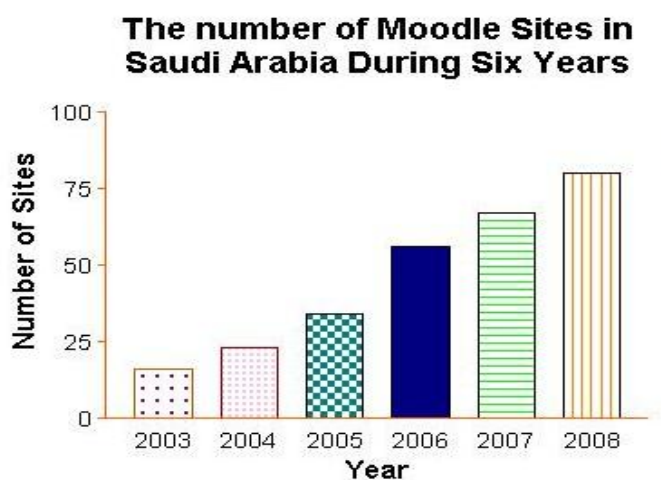


Figure 4.12: the increasing use of Moodle in Saudi Arabia over six years

This chapter has mentioned the VLE systems in Saudi Arabia. The use of Moodle was relatively limited in Saudi Arabia and there were just 16 sites in 2003, but now in 2008, more than 80 sites have registered with Moodle, as in Table 4.14.

Table 4.14: The Moodle Sites in Saudi Arabia in 2008.

No	Site	No	Site	No	Site
1	Saudi School in Leicester	25	Ajlan's High School	49	Conti Student VLE
2	بالمدينة المنورة / الجامعة الإسلامية عمادة التطوير الأكاديمي والإداري / الإلكتروني مركز التعليم	26	Electronic Board of General Studies Center التعليم الإلكتروني لمركز الدراسات العامة	50	College of Languages and Translation: E-learning system
3	etcdablom.net/moodle	27	والمعلومات قسم المكتبات	51	Conti Student VLE
4	A1 Integrated Cultural Services	28	Electrical and Computer Engineering	52	PSU Learning Management System (PSULMS)
5	isdept.info/moodle	29	BusyClass.com	53	e-learning @ Conti
6	Arab Open University	30	CNT English	54	English as a Second Language
7	Arab Open University LMS	31	عمادة تطوير المهارات	55	Faculty of Medicine
8	كلية علوم الحاسب و المعلومات	32	الدعم لتقنية المعلومات	56	FutureU Moodle
9	متوسطة العاصمة النموذجية	33	قائد الريم	57	كلية العلوم
10	Knowledge International University	34	Dept of English E- learning Courses	58	E-learning @ Jeddah Community College
11	الدكتور عبدالرحمن المحارقي	35	مجمع السلام التعليمي	59	Ibn Zahr Web Based Training
12	KSAALT	36	مدارس صفوة الفرسان	60	المكرمة الكلية التقنية بمكة
13	المنجبة للتعليم المطور	37	مدرسة الجبولوجيا	61	المركز التعليمي التقني الطائي (SLCT)
14	moodle.cc.edu.sa	38	مودل السعودية	62	Makkah E-learning
15	Online Collaborative writing	39	موقع المحاضر	63	أكاديمية نبض العلم
16	www.smartscool.com	40	موقع المدرب نبيل سنبيل	64	إدارة المقررات الإلكترونية
17	Open Gnu	41	نظام إدارة التعليم	65	البورد العربي
18	Open Gnu	42	الالكترونية حوزة الكوثر	66	التعليم الإلكتروني e learning
19	Optics Institute	43	Self-training	67	التعليم بمساعدة الحاسب
20	PSIS Online Class	44	SmartWay-School	68	مركز التدريب الإلكتروني لمعلمي الفيزياء
21	Aramco Moodle	45	Sooooqe.com	69	Collaborative Learning Center
22	Qassim College of Medicine	46	المدرسة الإلكترونية	70	Teaching Portal/Dr S. Nugali
23	Qassim University- college of Dentistry	47	Online Engineering Management Courses	71	A1 Integrated Cultural Services
24	Saudi Mentors	48	YIS Online Classroom	72	خالد جامعة الملك

4.7 Summary

This chapter has presented a comparison study between VLE features. According to a comparative study of the most well-known VLE systems, we have chosen Moodle as a suitable VLE system to consider it as a specific area of study. It is an important to make a comparison study between VLE products to select the suitable one and test it with our approach and also explore their strengths and limitations. This comparative study is in two phases. The first phase is based on the features and capabilities of VLE tools, and the second is based on the technical aspects of the systems of VLEs.

In this chapter, we have presented the literature on Moodle including its architecture, benefits, limitations, tools and the reasons for choosing this platform. Moodle is a kind of LMS and it is now widely used all over the world by schools, institutes, universities, companies and independent educators, and home-schooling parents. The software of Moodle allows users to download and use it on any computer device including web hosting, and it can scale from a single-teacher site to a more than 50,000-student-university site. Moodle has a large and diverse user community with over half a million registered users, speaking 75 languages in 193 countries.

In the next chapter, we will introduce the technical requirements for an SOC-based VLE (Moodle) by highlighting the requirements (functional and non-functional) for VLEs (Moodle) based on Web services. In the beginning of this chapter we will give a brief summary on distributed systems and distributed VLE systems. The main part of this chapter contains Web services for Moodle and the scenario of WSMS system, and its requirements for enabling Moodle to use Web services.

Chapter 5

Production of Technical Requirements for SOC Based on VLE (Moodle)

Objectives

- To introduce the technical requirements of SOC-based VLE (Moodle).
- To give a summary on distributed systems and the distributed system of VLE.
- To illustrate an example of Web services for Moodle.
- Requirements For VLEs (Moodle) based on Web services.

5.1 Introduction

Before determining an assignment program, we need to determine exactly what its technical requirements are. Technical requirements are a list of technical aspects that need to be satisfied before something is possible, such as requirements for running

Moodle. Here, the technical aspects of Moodle are of two types: functionality and non-functionality requirements.

Functionality requirements reflect the principle that architects should design, for example, a building based on the purpose of that building. They are a part of the answer to a question about why some object or process occurred in a system, which evolved through a process of selection or was designed with some goal in mind. On the other hand, non-functional requirements specify criteria that can be used to judge the operation of a system rather than specific behaviours, such as performance-related issues, reliability issues, and availability issues [127, 128].

This chapter briefly describes a production of technical requirements for an SOC-based VLE (Moodle) by highlighting the requirements (functional and non-functional) for VLEs (Moodle) based on Web services. At the beginning of this chapter we give a brief summary on the distribution systems of VLEs. Web services for Moodle are presented in Section 4. The main part of this chapter is technical requirements for VLEs (moodle) based on Web services including some examples.

5.2 Distributed System

Nowadays, Distributed Systems (DSs) are becoming more attractive both in industry and academia. The DS has evolved a great deal during the last three decades, and the spread of the Internet has made the area central for the development of modern software. DSs, such as grids, mobile agent systems, and large-scale clusters are gaining in popularity in every aspect of our lives. The development and testing of a DS is difficult and takes more time than the development and testing of Non-DS, which run on a single computer. In addition, a DS often has real-time constraints that increase the testing difficulty of such a platform [129, 130, 131].

A DS is a computer processing network where different parts of a program are

run simultaneously on one or more computers that communicate with each other. It contains a group of independent computers, connected through a network that allows these computers to organize their activities and share the resources over the network, thus browsers see the system as single, integrated computing. A DS works by dividing up one huge task into smaller pieces that can be processed at the same time but independently. It is the opposite of a centralized system because although a centralized one communicates with other computers, it appears to users as being a single, huge store of shared data, software and hardware [131, 132, 133, 134].

[131] defined Distributed System as:

‘A distributed computer system consists of multiple software components that are on multiple computers, but run as a single system. The computers that are in DS can be physically close together and connected by a local network, or they can be geographically distant and connected by a wide area network. DS can consist of any number of possible configurations, such as mainframes, personal computers, workstations, minicomputers, and so on. The goal of distributed computing is to make such a network work as a single computer’ [131, pp.13].

Today, DS is everywhere from WWW through interconnected banking systems to the network of computers in a modern aircraft. Therefore, this shift from single server environments to global DS presents a great challenge in terms of defining and enforcing appropriate computing policies. Nevertheless, providing good mechanisms for these systems has many advantages, and these are [129, 130, 131, 132, 133]:

1. **Sharing Resources:** the ability to use any hardware, software or data anywhere in the system.
2. **Performance:** using DS enables better performance instead of using single processor solutions.

3. *Fault Tolerance*: faults happen with hardware and software, and networks fail. DS can maintain availability even at low levels of hardware/software/network reliability. It is achieved through recovery and redundancy.
4. *Openness*: this is concerned with extensions and improvements of DS.
5. *Scalability*: the adaption of DS to accommodate more users and respond faster. The system can easily be expanded by adding more machines as needed.

As DS has many advantages, it also has some shortcomings and challenges that have faced developers and researchers in developing and deploying DS. These shortcomings and challenges are [129, 131, 132, 134]:

1. *Security*: private communications over public networks can allow others to observe the data and therefore change it. Database fragments need to be secured, and as DS is not centralized, remote sites must also be secured.
2. *Economics*: increased complexity and a more extensive infrastructure mean extra labour costs.
3. *Complexity*: engineers have the added burden of ensuring that the distributed nature of the system is transparent, and that they maintain multiple disparate systems, instead of one big one. In addition, they must build in extra database capacity to account for the disconnected nature of the database.
4. *Location, location, location*: placing data and computation for effective resource sharing, and finding it again once you put it somewhere.

5.3 The Distributed System of Virtual Learning Environments

The use of IT in learning and teaching in higher education institutions has increased significantly in recent years. Therefore, there are strong strategic drivers that VLE should become fully embedded into everyday learning and teaching practice, instead of only in traditional e-learning. This movement towards e-learning systems is not yet fully in place. So, the major challenges are to retain learners within education, and to encourage a wider group of learners to engage in higher education. One of the approaches that has been taken to meet these challenges is to use DS to encourage closer working and cooperation between higher education institutions [134, 135].

DS as a popular, essential, important and powerful computing paradigm will take a place in meeting these challenges and will support the large-scale use of learning systems. In parallel with and subsequent to DS, SOC as an example of DS has evolved with, for example, Web services [134, 135]. Our strategy, as in Figure 5.1, is aimed to extend the core idea behind VLE (Moodle) tools, which are gradually dominating academic institutes. The fundamental contribution that we are aiming for is to build a VLE (Moodle) around Web services concepts.

Web services, as an example of SOC, support the integration of software applications in an incremental way, using existing platforms and languages that utilize and adopt existing legacy systems. VLEs are one of the emerging domains that require to be extended to SOC. Web services techniques such as service composition, descriptions, registrations, discovery and binding has opened a gate for collaborative VLE services that are flexible, widely distributed and effective. In order to prove the concept, we have selected a service (courses and assignments), provided by a well-known VLE product (Moodle).

5.4 An Example: Web Services for Moodle

Nowadays, the e-learning environment has more than 250 commercial e-learning platforms, and more than 45 OSS offerings, as in the previous chapter. Of these latter ones, the more well known are Moodle, Ilias, Eduplone, Claroline, SAKAI, WebCT and Bscw; they have wide developer communities and present strong arguments for considering open source a straight-forward and potentially feasible competitor to commercial products. One open-source project of particular note, which has emerged to meet the growing interest in open-source platforms, is Moodle [83, 85, 86].

Moodle is a web based CMS and VLE designed around pedagogical principles, namely a social constructivist philosophy using the collaborative possibilities of the Internet [109]. It lets teachers provide and share documents, graded assignments, quizzes, discussion forums, etc. with their students in an easy-to-learn manner and to create quality online courses. Moodle is a free OSS, which means users are free to download, use, modify and even distribute it under the terms of GNU [107, 83, 110].

The goals of Web services are to standardize the roles, terminology, concepts, message exchanges and the WSDL needed to express the notification components, and to provide a language to describe topics. In meeting these goals, the Web services specifications must explicitly address the functional requirements of its customers. Therefore, it is important that a Web services framework is augmented so that the functional requirements of a Web service can be determined at run-time, and so that customers are bound to a service that best meets their functional requirements as well as their non-functional requirements, which we will describe at the end of this chapter.

This study proposes Web Services Matching and Selection (WSMS) system as an extension of the existing Web services framework, which enables a collection of functional and non-functional service requirements at run-time, the usage of col-

lected data in discovery, binding, and the execution of Web services, as below in Figure 5.1.

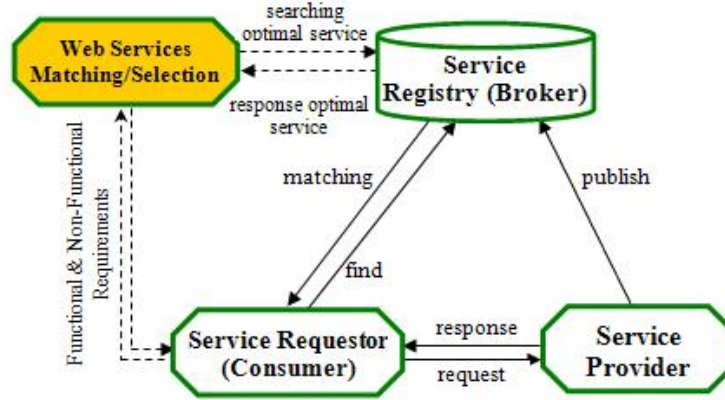


Figure 5.1: The Proposed Architectural Extension Based on Web Services for VLE

WSMS as in the top left of Figure 5.1 gives more flexibility to controlling the connecting operation and obtaining the optimal service between consumer and services. The most important advantage of this system is that it reduces the selection services by using the attributes of both consumer and service, and it then offers the optimal one to the consumer. Thus, this system is a friendly and easily accessible mechanism for the discovery and selecting of services, and thereby has built trust between the consumer and services communities. In addition, this section proposes WSMS based on the UML modelling approach for Web service requests and offers, as below in Figure 5.2 [136].

WSMS system suggests the new extension of Web services structural design, which keeps data around the collection of Web services and frequently explores these services to collect data about their functional and non-functional requirements. WSMS system collects and analyses information to deduct additional requirements of the service. Some of the major modules of WSMS system and their functions and requirements are described below in brief, and this system will be explained more deeply in Chapter 7 [136].

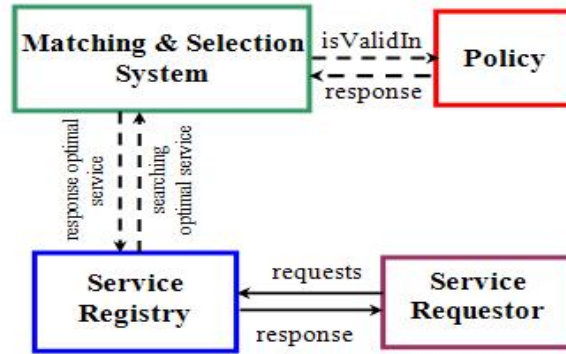


Figure 5.2: UML for representing Web service offers and requests

This system starts with the discovery of a service by a potential user, and a correspondence is established between the objectives of the consumer and the capabilities of the service. An application does not need to discover all the available services matching a set of requirements and specifications, it just needs to find one that is good enough in terms of features and quality. By basing discovery on meeting a specified quality of service requirements, we decrease irrelevant outcomes, thereby improving the payoff of the discovery computation[4].

WSMS system has used the environment of ponder language policy, which provides a common means of specifying security policies that map onto various access control implementation mechanisms for firewalls, operating systems, databases and Java. In addition, it supports obligation policies that are event-triggered condition-action rules for the policy-based management of networks and distributed systems. Ponder can also be used for security management activities such as registration of users or logging and auditing events for dealing with access to critical resources or security violations [137]. Chapter 7 has more information about the Ponder policy and how WSMS applies this policy to be suitable for Moodle based on Web services.

5.5 Requirements for Virtual Learning Environments (Moodle) Based on Web Services

Web services are applications that allow for requests from other systems across the Web through vendor-neutral communication technologies, and that enable any network to interact. Web services began by evolving and operating in an exceedingly intelligent and dynamic way; they comprehend the context of every request and create results based on each particular situation. Web services perform their processes based on the user's location, identity, reason, and preferences for a request[138].

This section highlights the functional and non-functional requirements of course and assignment activities in a Moodle system. Certain features are taken from course and assignment tools and are integrated into Moodle activities that allow for the integration and upload of a single file, online text or offline assignment. They also allow advanced uploading of files and resubmitting, determine maximum sizes, allow comment inline and permit non-standard assignment types.

5.5.1 Functional Requirements

A functional requirement is the principle that an architect should design a project based on the purpose of that project. They are a central aspect of the reasons why some objects and processes occur in a system that has evolved through a process of selection and was designed for a particular targets. This requirement indicates functionality in specific bulleted items about what exactly a certain service accomplishes. Service operations, methods, actions etc, must be defined in terms of what part of the functionality it provides [127, 128].

As in the top left of Figure 5.3, WSMS involves discovering a set of semantically equivalent services by filtering a number of available services based on service meta-data, and selecting an optimal service based on real-time during previous executions

of a service. WSMS has some functional requirements that enable a user to find the optimal service, and these requirements are related to Web services using some main elements and attributes. These functional requirements are as below:

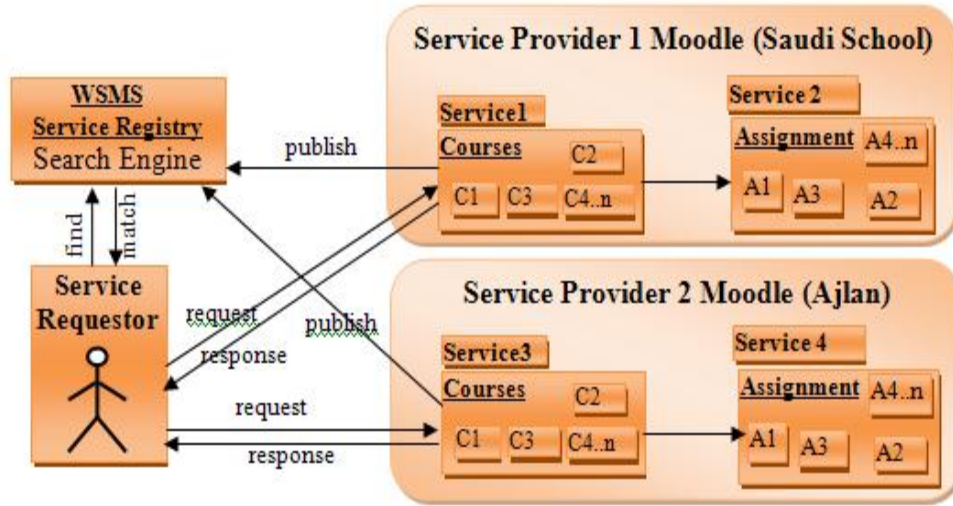


Figure 5.3: Web Services in Multi Moodle Platforms

5.5.1.1. ComplexType

WSMS not only receives basic data types and returns single values, but it can also create complex data types to allow different structures to be passed to and from the server. In this section, a more complex data type is created to return more details about courses and assignments in two schools (Ajlan's High School (AHS) and Saudi School in Leicester (SSiL)). We have in every school many complexTypes such as Course, Assignment, AssignmentGradesInput, AssignmentGrade, Student, UserDataInput, UserDataOutput, and UserCounts. This section will focus on two complex data type; the first is for Course and the second is for Assignment, as follows:

5.5.1.1.1. ComplexType for Course

Complex data type for course has an array that is a data structure consisting of a group of elements that are accessed by indexing. Each element has the same data type and the array occupies a contiguous area of storage. Most languages array a complex data type so as to return more and complex details [138].

Variables can store a series of related values by using an array, and in some situations store a single value. Figure 5.4 shows the course details that have been chosen from both schools (AHS and SSiL). The course has many details and we have chosen the necessary information, which is id, full name, summary, start_date, assignments_count, students_count, category and school.

```
$server->wsdl->addComplexType(
    'Course',
    'complexType',
    'struct',
    'all',
    '',
    array(
        'id' => array('name' => 'id', 'type' => 'xsd:int'),
        'fullname' => array('name' => 'fullname', 'type' => 'xsd:string'),
        'summary' => array('name' => 'summary', 'type' => 'xsd:string'),
        'category' => array('name' => 'category', 'type' => 'xsd:string'),
        'start_date' => array('name' => 'start_date', 'type' => 'xsd:dateTime'),
        'assignments_count' => array('name' => 'assignments_count', 'type' => 'xsd:int'),
        'students_count' => array('name' => 'students_count', 'type' => 'xsd:int'),
        'school' => array('name' => 'school', 'type' => 'xsd:string')
    )
);
```

Figure 5.4: Complex Data Type for Course

5.5.1.1.2. ComplexType for Assignment

Assignment activity is what one activates under course, and which gives both teacher and student a flexible way to achieve homework successfully. Figure 5.5 shows the assignment details from both schools (AHS and SSiL). The assignment activity has many details and we have chosen the necessary information, which is ID, name, description, assignment type, resubmit and grade.

```
$server->wsdl->addComplexType(
    'Assignment',
    'complexType',
    'struct',
    'all',
    '',
    array(
        'id' => array('name' => 'id', 'type' => 'xsd:int'),
        'name' => array('name' => 'name', 'type' => 'xsd:string'),
        'description' => array('name' => 'description', 'type' => 'xsd:string'),
        'assignmenttype' => array('name' => 'assignmenttype', 'type' => 'xsd:string'),
        'resubmit' => array('name' => 'resubmit', 'type' => 'xsd:int'),
        'grade' => array('name' => 'grade', 'type' => 'xsd:int'),
        'school' => array('name' => 'school', 'type' => 'xsd:string')
    )
);
```

Figure 5.5: Complex Data Type for Assignment

The course and the assignment, as in Figure 5.5 and 5.4, are the names of the complex data type. The array defines all the elements that will combine to form the course and assignment type. In this case, each of the items is a basic type, but we can nest complex types within each other, or use more advanced types like arrays.

5.5.1.2. Register Service

The second requirement works to register the functions in the server class in NuSOAP, and we have until now 9 functions as in Figure 8.7. Figure 5.6 shows the registration of the getCourses function and the other functions are in Appendix B1.

```
$server->register('getCourses',      // method name
    array(),
    array('return' => 'tns:ArrayOfCourses'),
    'serverwsdl',                    // namespace
    'serverwsdl#getCourses',        // soapaction
    'rpc',                          // style
    'encoded',                      // use
    'Get all current courses'       // documentation
);
```

Figure 5.6: Register getCourses Services in Server

5.5.1.3. Data Source

The data source requirement is responsible for connecting with the database and obtaining the required data for the course and assignment module. This service has 9 functions as in Figure 8.7. This section has only one function that connects with the databases, as in Figure 5.7, which is for AHS.

```
function getCourseDetails($courseid) {
    mysql_connect('localhost','aj_lal','gLv13');
    mysql_select_db('ajlanal_mdle1');
    $query = " SELECT cs.id, cs.fullname as course, cs.summary as summary, cg.name as category, "
        . " FROM_UNIXTIME(cs.startdate) AS start_date, "
        . " (SELECT Count(a.id) FROM mdl_assignment a " //Get Assignments Count
        . " INNER JOIN mdl_course c ON a.course = c.id AND c.category <> 0 "
        . " INNER JOIN mdl_course_categories cg on cg.id = c.category "
        . " Where c.id = cs.id "
        . " ) as assignments_count, "
        . " (SELECT Count(Distinct ra.userid) FROM mdl_user u " //Get Students Count
        . " INNER JOIN mdl_role_assignments ra ON u.id = ra.userid "
        . " INNER JOIN mdl_context con ON ra.contextid = con.id AND con.contextlevel= 50 "
        . " INNER JOIN mdl_course c ON con.instanceid = c.id AND c.category <> 0 "
        . " INNER JOIN mdl_course_categories cg ON c.category = cg.id "
        . " INNER JOIN mdl_role r ON ra.roleid = r.id AND r.name = 'Student' "
        . " Where c.id = cs.id "
        . " ) as students_count "
        . " FROM mdl_course cs "
        . " Left Outer Join mdl_course_categories cg on cg.id = cs.category AND cs.category <> 0 "
        . " Where cs.id = '$courseid' "
        . " ORDER BY cs.fullname, cg.name, cs.startdate "
        . " LIMIT 1 ";
    mysql_query("SET NAMES 'UTF8'"); //So important for Arabic Text
    $result = mysql_query($query) or die(mysql_error());
    $num_rows = mysql_num_rows($result);
    if ($num_rows > 0)
    {
        $row = mysql_fetch_array($result);
        $tempCourseObject['id'] = $row['id'];
        $tempCourseObject['fullname'] = $row['course'];
        $tempCourseObject['summary'] = $row['summary'];
        $tempCourseObject['category'] = $row['category'];
        $tempCourseObject['start_date'] = $row['start_date'];
        $tempCourseObject['assignments_count'] = $row['assignments_count'];
        $tempCourseObject['students_count'] = $row['students_count'];
        $tempCourseObject['school'] = getSchoolName();
    }
    else
    {
        unset($tempCourseObject);
    }
    return $tempCourseObject;
}
```

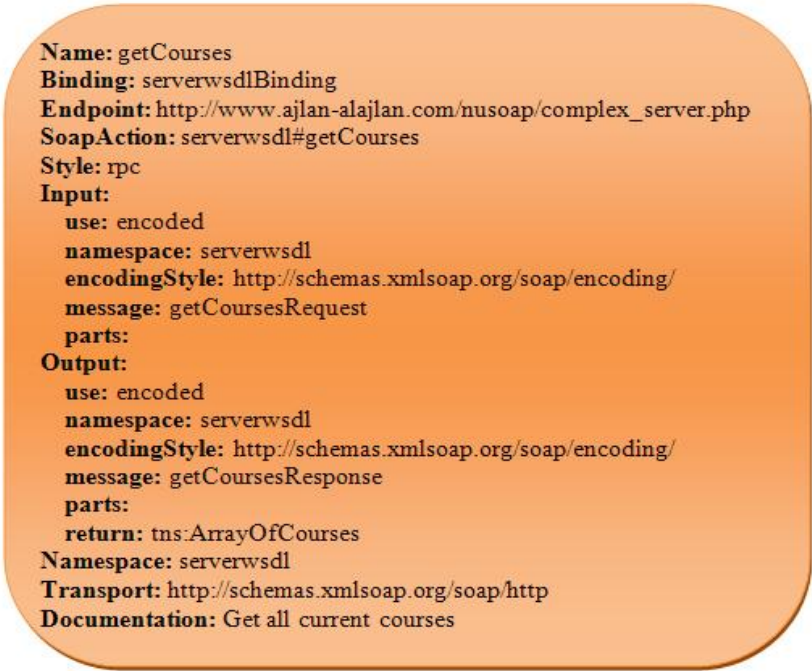
Figure 5.7: Courses connects with database (SQL)

5.5.1.4. NuSOAP

NuSOAP is an important requirement that enable us to generate WSDL files for services by registering our functions in the NuSOAP server. Introducing Web ser-

vices to Moodle is one of the main objectives of our study, and recently, Moodle developers have been facilitating the discussion about the development of Web services. The best way to deliver Web services to Moodle is through the NuSOAP approach. Allowing the NuSOAP package to dynamically generate applications can help a great deal with debugging or adding new service methods. NuSOAP is a good choice for creating and consuming PHP SOAP services for reasons that will be explained in Chapter 8.

NuSOAP works as an x-UDDI registry and generates WSDL files. Therefore, it provides information about every service that has registered with it. For example, the `getCourses` function has the details, as in Figure 5.8, when registered with the NuSOAP server. For more information about other functions and their WSDL, please visit http://ajlan-alajlan.com/nusoap/complex_server.php#, as well as Chapter 8 has more details on how NuSOAP generates WSDL files.



```
Name: getCourses  
Binding: serverwsdlBinding  
Endpoint: http://www.ajlan-alajlan.com/nusoap/complex_server.php  
SoapAction: serverwsdl#getCourses  
Style: rpc  
Input:  
  use: encoded  
  namespace: serverwsdl  
  encodingStyle: http://schemas.xmlsoap.org/soap/encoding/  
  message: getCoursesRequest  
  parts:  
Output:  
  use: encoded  
  namespace: serverwsdl  
  encodingStyle: http://schemas.xmlsoap.org/soap/encoding/  
  message: getCoursesResponse  
  parts:  
  return: tns:ArrayOfCourses  
Namespace: serverwsdl  
Transport: http://schemas.xmlsoap.org/soap/http  
Documentation: Get all current courses
```

Figure 5.8: The details of the `getCourses` service after registering with NuSOAP server

5.5.1.5. Matching and Selection System

As in the top left in Figure 5.3, WSMS gives the user the ability and flexibility to search in the service registry, and to match all services that meet user needs. The important advantage of this system is that it reduces the selection service by using some attributes of both consumer and service in order to offer the optimal one to consumer. These attributes depend on a few requirements: location, name of course, name of school, number of services and time.

Selection mechanism is part of the WSMS system, which works with policy language to state and find the optimal service, and by offering only the best services, a user can select the optimal one; the selection that best matches his/her requirements. After the user has found the services, the binding services are enacted, which means how the user connects to, and interacts with, a suitable Web service after it has been found in the service registry. Binding happens between the service provider and the service requestor when the user finds his/her service from the service registry.

5.5.2 Non-Functional Requirements

Non-functional requirements specify criteria that can be used to judge the operation of a system, rather than specific behaviours, and these usually include flexibility, security, trust, changeability, extensibility, reliability, availability and accessibility issues. These non-functional requirements come into play when Web services are constructed and deployed. This section details the non-functional requirements that are required when both schools (AHS and SSiL) use Web services, and these requirements are:

5.5.2.1. Flexibility

Flexibility has become one of the more strategically competitive tools and is usually a prerequisite. It refers to the availability of alternative resources. These resources

may have diverse parameters, especially those related to physical and operating systems. Typically, flexibility is one of the major advantages of using Web services, which are a set of protocols based on XML/SOAP over HTTP. The service is accessible using SOAP, which offers more flexibility to developers because they can use the technology that they are most comfortable with [139].

Web services and related techniques have opened a wide door for e-learning services that are flexible, widely distributed and effective. WSMS, as an extension of Web services, offers the most flexibility and functionality by using the features of Web services. It provides teachers much more flexibility than traditional systems because it allows educators at different institutions in different countries to work together and share material by connecting individual courses together, which are hosted on different VLEs. Therefore, 10 educators at different institutions that teach the same course and can share assignments, chat, etc, and communicate with each other. At the same time, WSMS gives all users more flexibility by availing them many advantages and some of them as below and the others in Chapter 8:

- To search in all schools (here, AHS and SSiL) that register with WSMS.
- To display all courses and assignments in all schools by school and other categories in order to help users to choose the courses that best suit their needs.
- User can search by date (he/she can specify period and get the courses or assignment or both in this period
- User can search by teacher.

5.5.2.2. Security

Security is a serious issue that every developer must be concerned with when writing applications. It is always a concern for developers and becomes more of an issue the more publicly available an application is. It is no different when working with XML.

In many cases, it becomes more of an issue because XML documents are sent to and received from remote sources. It is therefore important that the data can be trusted before being processed. Security specifies how well a service provides confidentiality and non-repudiation by authenticating the parties involved, encrypting messages, and maintaining access control. A service provider can provide different security levels and mechanisms depending on the service requester[140]. WSMS security framework has to enable the following points [25, 140, 138]:

5.5.2.2.1. Authentication

Authentication is for the benefit of administrators, teachers, students and guests. WSMS knows the parties that are already logged into the system and enables them to access their school directly without logging in every time. It enables users to obtain the data they need, such as names of the schools that register with WSMS, number of courses in each school, activities, teachers and students, and so on.

Persistent and transient authentication of authorship of data is necessary for important software packages, especially for banks and e-learning platforms. Thus, WSMS has the ability to identify users and give them permission for their security level, as in Figure 5.9.

5.5.2.2.2. Authorization

The authorization process is used to decide if user, device and program is allowed to have access to data or service. Authorization is the principle of enabling access to computer resources only for administrator, teacher, student and guest. It is a process that defends computer resources by just enabling those resources to be used by resource clients that have been granted authority to use them.

The authorization in WSMS is divided into five levels: administrator for WSMS, administrator for schools, teacher, student and guest. As in Figure 5.9, every level

has some permission that WSMS has permitted to their users. For example, an administrator for WSMS has 100% access and can do anything in the WSMS site, but an administrator for schools has only 75% access although he/she also has 100% access to related schools and can do anything there, as in Figure ?? in Chapter 4.

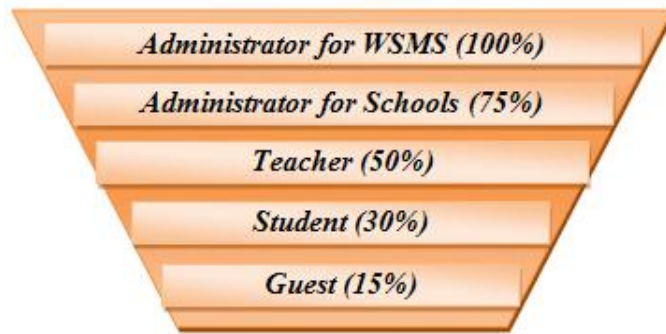


Figure 5.9: The Levels of Authorisation in WSMS

5.5.2.2.3. Confidentiality

Confidentiality is ensuring that information is accessible only to those authorized to have access, as in Figure 5.9. WSMS has provided confidentiality for its users by giving them the ability to access with same login details. Every user can see the information about all schools, but he/she cannot necessarily access those schools; users can access their own school as a normal user in their school.

5.5.2.3. Trust

Trust is confident reliance. We may have confidence in people, events, or at least in our beliefs, but if we do not rely on them, our confidence alone does not amount to trust. This requirement determines the specifications for secure messaging mechanisms to facilitate trust relationships in diverse Web service environments. For a secure communication between two parties, both parties must exchange security credentials using a specification.

Using the basic SOAP, WSDL, UDDI and HTTP framework for Web services, users can engage in secure communications. When a request containing security tokens arrives at a service as a SOAP message, Web services trust requires that the message prove a set of claims, such as name, key, and permission. Otherwise, the service will deny the request or ignore the message. To validate a request, the service must have a trust engine that [4]:

- Verifies that the claims match the requirements of the service's policy,
- Verifies that the signature proves the attributes of the claimant, and
- Verifies that the security tokens are trusted to issue their claims. As an alternative, a trusted intermediary may verify the requester by simply asserting the requester's identity.

WSMS must have security that is sufficiently strong enough for all schools around the world to trust it. Exchanging data over the Internet can be dangerous and prone to fraud or being lost. Therefore, we are increasing our efforts to combat fraud on the system with a variety of programs. We are also making sure our disclosures about cooperating with law enforcement bodies allow us to respond to increasingly sophisticated fraudsters and criminals. We are using messages to authenticate users if, for example, they are using an unfamiliar computer in order to prevent fraudsters from abusing users' accounts.

5.5.2.4. Changeability

The Web is moving from being a collection of pages toward a collection of services that interoperate through the Internet. Therefore, changeability is necessary for the existing systems but handling change is a fundamental problem in software engineering. Evolutionary development has been proposed as an efficient way of dealing with risks such as new technology and imprecise or changing requirements. Therefore,

WSMS resolved risk at an early stage by incrementally evolving the system towards completion instead of relying on the traditional approach. It designs and maintains open-ended architecture for the success of evolutionary software engineering processes [141].

WSMS is open source and free, and this enables developers to make changes under the terms of the GNU license. This has engendered a high level of trust for developers so that they can update and build new technologies. It has also given schools good reason to trust this system and to publish services, especially their secure services such as grades.

5.5.2.5. Extensibility

Extensibility is the ability to add new features to an existing program without disturbing any existing code. Extensibility, as in [4], is:

'It is a system design principle where the implementation takes into consideration future growth. It is a systemic measure of the ability to extend a system and the level of effort required to implement the extension. Extensions can be through the addition of new functionality or through modification of existing functionality' [4, pp.111, 434].

WSMS has the ability to extend and add more features in the future. Using the basic SOAP, WSDL, UDDI and HTTP frameworks for Web services makes extensibility easier and allows for new functions and modifications to be added to the existing functionality. For example, houses can be built with future extensions in mind to enable the owner or a new owner to make the extension easily without destroying any other parts. Currently, WSMS deals with two schools and both are using the Moodle platform, which is OSS. These kinds of platforms enable users to freely download, use, modify and even distribute them under the terms of GNU.

5.5.2.6. Reliability

Reliability is an important requirement, and is the ability of a system to perform and maintain its functions in routine circumstances, as well as in unexpected circumstances. Reliability measures how often a service meets or exceeds its advertised capabilities, typically specified as inversely proportional to the service's number of failures per time interval [4, 140].

The IEEE defines it as:

‘Reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time’.

5.5.2.7. Availability

Availability is a measure of whether a service is present and ready for immediate use, or likely to be ready when it is needed. It can be specified as the probability that the service is available, or specified in terms of the particular times it is usable or, alternatively, the times when it is under repair [4, 138].

For example, a school might update its courses and assignments and the service might be unavailable during this time. The WSMS system cannot match with services that are not available in the databases of schools. As in Figure 5.7, the function `getCourses` can only connect with a database that is available if it is to obtain all the courses in AHS.

The advantage of WSMS is that it immediately makes a course or assignment service available when a teacher creates a course or assignment. This enables users to discover new services at once in both schools (AHS or SSiL).

5.5.2.8. Accessibility

Accessibility is a quality measure representing the likelihood that a service can satisfy a request at a given point in time. A service might be available but inaccessible due

to network traffic or busy. For example, a Web service submitting assignments to deadlines for a large university might be inaccessible because of a high demand from student applications trying to submit assignments. WSMS has a high accessibility, which it achieves by moving users to the nearest service or to the server does not have a high demand from students. Building scalable systems that can serve requests consistently despite variations in volume is the target of WSMS to be sure that users can get their requests without any delay or unavailable service [140].

5.6 Summary

This chapter has presented the key points about the production of the technical requirements for SOC based on VLE (Moodle). Functionality requirements are the principle that architects should design something only based on its purpose. They are part of the answer to the question about why some object or process occurs in a system that has evolved through a process of selection or was designed for a specific reason. On the other hand, non-functional requirements specify criteria that can be used to judge the operation of a system, rather than specific behaviours, and these typically include performance-related, reliability, and availability issues.

This chapter has described the technical requirements for Web services based on VLE (Moodle). It focused on a course and an assignment services in Moodle and on how to use its concepts to help users informally identify technical requirements or talk with others about technical requirements using standard terminology.

The next chapter will design the architecture for a SOC based on VLE. It will fulfil the demand for a suitable syntax for SOC based on VLE by sketching a UML for SOC with an example. It adapts the UML metamodel as defined in the specification document of a typical SOC. In addition, the next chapter will present the scenario of Web services for WSMS and use JFLAP with FSM to design the states of WSMS.

Chapter 6

The Architecture for an SOC-Based VLE (Moodle)

Objectives

- To present the architecture for an SOC-based VLE (Moodle).
- To describe the role of the SOC approach, displayed in a UML package.
- To examine matching for specific types by UML language.
- To present the scenario of Web services for WSMS.
- To use JFLAP with FSM to design the states of WSMS.

6.1 Introduction

Modelling has been an active area in software development over recent years. When writing applications, from using the simplest languages to the most powerful and complex languages, there is still the need to model. Defining a model makes it easier to break up a complex application or huge system into simple, discrete pieces that

can be individually studied. A model allows developers to focus more easily on the smaller parts of a system and then to understand the whole picture.

Currently, services are one of the most important issues within the scope of Web Information Systems (WIS) development. Therefore, there are now a number of middleware platforms that allow the implementation of Web services and that facilitate the service-oriented applications development. However, the need for a solid methodological base for the development of Web services and service-oriented applications has increased the need for modelling techniques and methods that can guarantee quality in the development of these sorts of applications [68, 142]. The importance of a Unified Modelling Language (UML) for data and process modelling in SOC projects has been underplayed. It adapts the UML metamodel as defined in the specification document for typical SOC [143, 144].

This chapter designs the architecture for a SOC based on VLE (Moodle). It will fulfil the demand for a suitable syntax for SOC based on VLE (Moodle) by sketching a UML for SOC with an example. It adapts the UML meta-model as defined in the specification document of a typical SOC. In addition, this chapter presents the scenario of Web services for WSMS and uses JFLAP with FSM to design different WSMS states.

6.2 Unified Modeling Language

UML is a standard language for constructing, visualizing, specifying, documenting the artefacts of software systems, and for business modelling and other non-software systems. UML is a notation, which is graphical language with rules for creating analysis and design methods. UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

UML is a very important part of developing object-oriented software and the

software development process. UML uses mostly graphical notations to express the design of software projects. Using UML helps scheme teams communicate, explore potential designs, and validate the architectural design of the software. UML is a supporting tool for the project and design process, and involves the creation of various graphical or text based documents [127].

6.2.1 History of Unified Modelling Language

UML began to appear between the mid-1970s and the late 1980s as a variety of methodologists experimented with different approaches to object-oriented design and analysis. The number of identified modelling languages increased from less than 10 to more than 50 during the period 1989 to 1994. In the middle of 1990s, new iterations of these methods began to emerge and these methods began to incorporate each other's techniques, and a few prominent methods came out.

The development of UML began in the end of 1994 when the primary authors, G. Booch and J. Rumbaugh of Rational Software Corporation (RSC), began their work on unifying the Booch and Object Modelling Technique (OMT) methods. In 1995, I. Jacobson and his Objectory Company joined Rational to unify their efforts, merging them with the Object-Oriented Software Engineering (OOSE) method. In the middle of 1996, the attempts of Booch, Rumbaugh, and Jacobson resulted in the release of the UML 0.9 and 0.91 documents.

IBM, ObjecTime, Ptech, Taskon, Softeam, Reich Technologies and Platinum Technology joined the UML partners to contribute to their ideas, and together they produced the revised UML 1.1 in the beginning of 1997. The focus of the UML 1.1 release was to improve the clarity of the UML 1.0 semantics and to incorporate contributions from the new partners. It was submitted to the OMG for their consideration and adopted in the autumn of 1997 [145, 146].

6.2.2 The Aims of UML

UML has provided developers with many advantages that have made programming easier [142, 146, 147], and these are that UML:

1. Offers easier understanding of a system, which is the first step in either building or enhancing a system.
2. Supplies users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models.
3. Provides extensibility and specialization mechanisms to extend the core concepts.
4. Is independent of particular programming languages and development processes.
5. Encourages the growth of the OO tools market.
6. Supports higher-level development concepts such as collaborations, frameworks, patterns and components.
7. Offers many different notations to represent dynamic behaviour, such as state chart diagrams, sequence and collaboration diagrams.
8. Supports the principle of separation of views, which is an effective means of controlling complexity.

6.2.3 Why Use UML?

Developers and researchers use UML language for many reasons and here we will mention some of these reasons as follows [142, 145, 146, 147]:

1. Software is very abstract and hard to visualize

A visual modelling language such as UML allows software to be visualized in multiple dimensions so that software can be completely understood before construction begins. Furthermore, UML can be used to produce several models at increasing levels of detail.

2. The production of software and to improve quality

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market.

3. Appropriate for both new and legacy systems

UML is appropriate for both new system developments and for improvements to existing systems. It is a fallacy that in order to use a new modelling technique on an old system, the old system will have to be completely documented in the new style in order for any change to take place.

4. Solve recurring architectural problems

Companies recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. UML was designed to respond to these needs.

5. Accommodates incremental development and re-development

UML responds well to an incremental development environment. With UML it is possible not only to develop just the parts of the model that are required to satisfy the new requirements, but also to demonstrate that all the codes needed to fulfil those requirements are in place.

6.3 The Architecture for an SOC-based VLE (Moodle)

A general scheme of the e-learning system for trend modelling is here presented. Efficient learning is a requirement for students in many subjects. Therefore, further development of VLE is required, especially distributed applications, and these applications need to be supported by platforms such as Web services. The aspects of the technology shared by these platforms, such as the roles of service providers, service requesters, and service registries as well as their publish, find, and bind operations, are conceptualized in SOC [148].

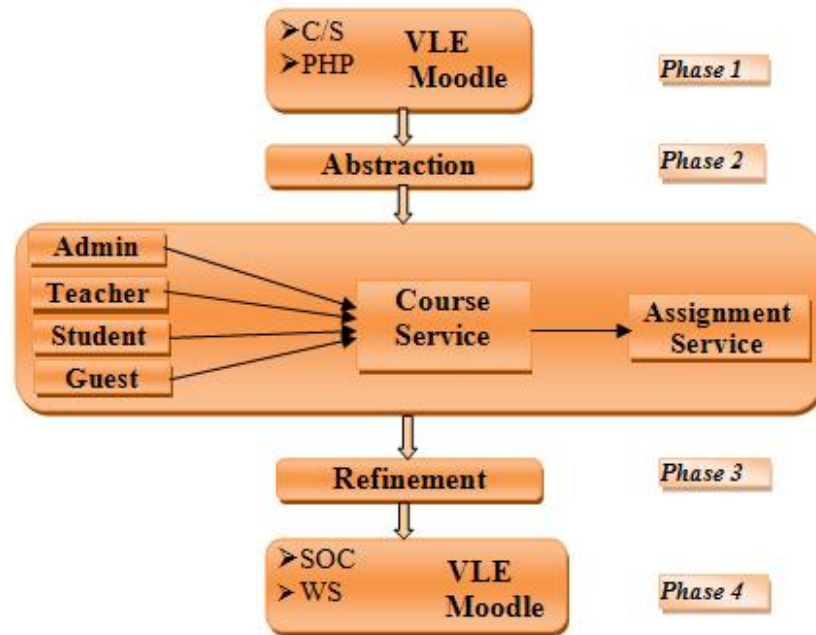


Figure 6.1: The Architecture for an SOC-based VLE (Moodle)

In this section, we discuss how VLE (Moodle) can use a new technology, which is the Web services. The idea of this technology is that Moodle will move from its current position, which is Client/Server (C/S), to a new position, which is Web services technology. The model moved the current Moodle from Web application to applying Web services. As in Figure 6.1, this model has four phases as follows:

6.3.1. Phase I

C/S is a network architecture that separates a client from a server by using a Graphical User Interface (GUI). Each instance of the client software can send requests to a server. Specific types of servers include: application servers, file servers, terminal servers, and mail servers. Currently, the C/S model is still used on the Web, such as by Moodle, where a user may connect to a service operating on a remote system through the Internet protocol suite. Web browsers are clients that connect to Web servers and retrieve web pages for display.

Moodle depends on C/S, which is based on PHP language, and all content of operations and codes are in the server and the client, just as with web pages, are passed on in html. The browser requests the application that is on the server, as in Figure 6.2. Here, we will focus on the server side, which has two parts:

1. Presentation logic: The processing (instructions, routines, etc.) required to present data. In general, it refers to the execution of GUI such as html. Presentation logic means how we display these objects to a browser.
2. Business logic: The piece of an application program that performs the required data processing of the business, such as PHP [149].

Sometimes, it is difficult to determine where business logic ends and presentation logic begins. Occasionally, the business and presentation logics are so intricate and interdependent that they become intertwined and can be difficult to separate them.

Why should we separate Presentation Logic from Business Logic?

There are a number of reasons why separating presentation logic from business logic is preferable, and below are the most important ones [149, 150]:

1. This technique enables the user to improve performance by optimizing the parts separately, and allows the user to reuse the business logic with different

forms of presentation.

2. Presentation logic and business logic do not occur at the same time. Separating them minimizes time dependency and allows for concurrent development. Developers can work on each part independently and then integrate them later.
3. They allow separation of workforce, which is important for large organizations with very specialized developers.

In general, it is good practice to split applications into one part containing presentation and business logic that is reusable, and one part responsible for presentation to the client as in Figure 6.2.

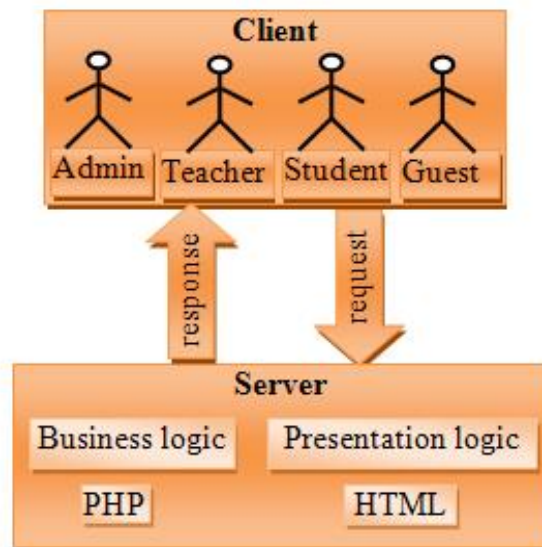


Figure 6.2: Client/Server in VLE (Moodle)

In the current Moodle, the separation of presentation and business logic is not clear, as can be seen in Figure 6.3 and 6.4. They are in the same file (Moodle/question/preview.php) and their position is in the server. As in Figure 6.3, presentation logic means how we display these objects to a browser by using the HTML language. In Moodle, presentation logic is in the server with business logic, and therefore presentation logics should move to the client to facilitate the process and

improve performance as well as to reduce time dependency and enable concurrent development. Researchers are able to work on each part independently and then put them together later.

Presentation Logic (Moodle)

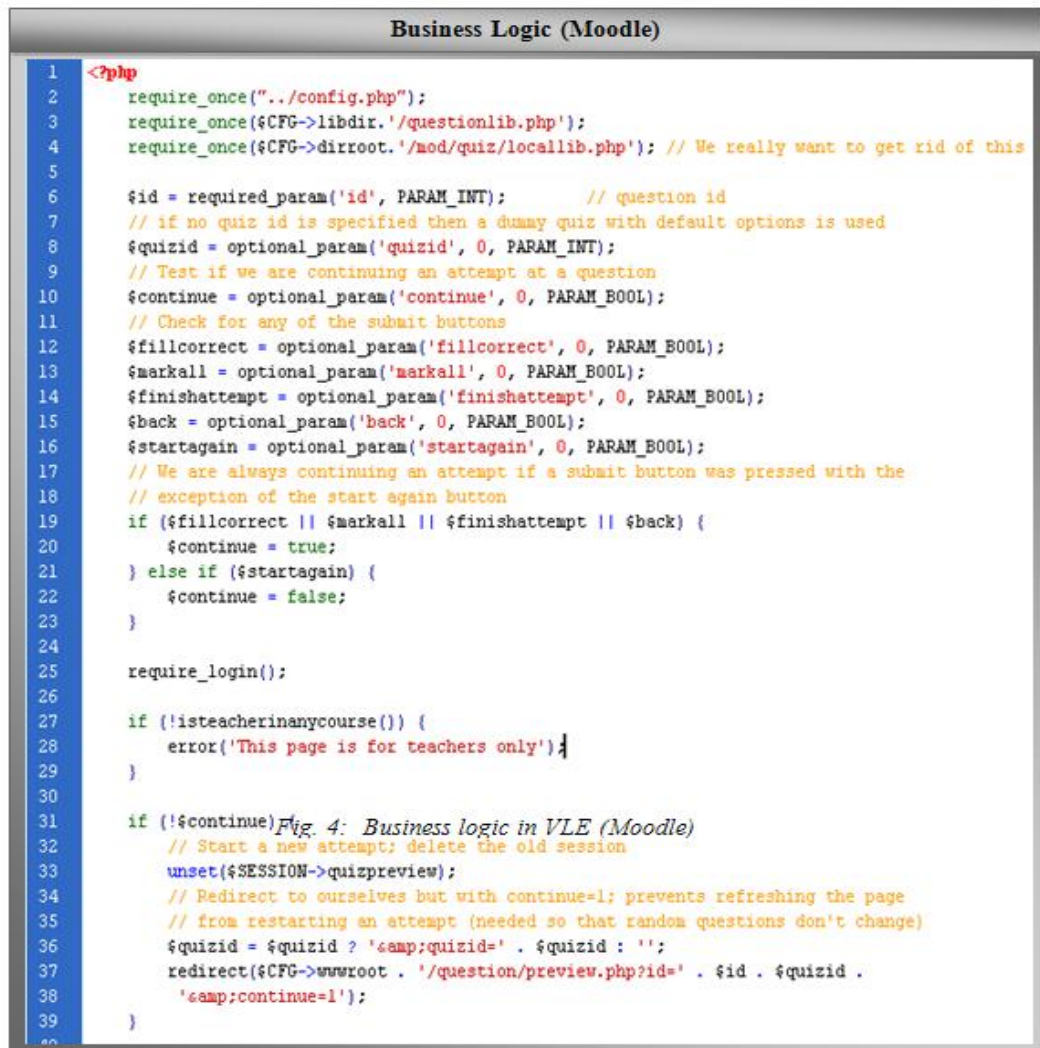
```

197     echo '<br />';
198     echo '<center>';
199
200     // Print the mark and finish attempt buttons
201     echo '<input name="markall" type="submit" value="' . get_string('markall',
202         'quiz') . "\" />\n";
203     echo '<input name="finishattempt" type="submit" value="' .
204         get_string('finishattempt', 'quiz') . "\" />\n";
205     echo '<br />';
206     echo '<br />';
207     // Print the fill correct button (unless the question is in readonly mode)
208     if (!$options->readonly) {
209         echo '<input name="fillcorrect" type="submit" value="' .
210             get_string('fillcorrect', 'quiz') . "\" />\n";
211     }
212     // Print the navigation buttons
213     if ($historylength > 0) {
214         echo '<input name="back" type="submit" value="' . get_string('previous',
215             'quiz') . "\" />\n";
216     }
217     // Print the start again button
218     echo '<input name="startagain" type="submit" value="' .
219         get_string('startagain', 'quiz') . "\" />\n";
220     // Print the close window button
221     echo '<input type="button" onclick="window.close()" value="' .
222         get_string('closepreview', 'quiz') . "\" />";
223     echo '</center>';
224     echo '</form>';
225     print_footer();
226     ?>

```

Figure 6.3: Presentation logic in VLE (Moodle)

Figure 6.4 displays the business logic, which has a PHP code, and is in the server with presentation logic. It is the piece of an application program that achieves the required data processing of the business, in PHP code, as in Figure 6.4. As we said in presentation logic, separating presentation logic and business logic will give Moodle and other VLE products the advantages that we mentioned above.



```

1 <?php
2 require_once("../config.php");
3 require_once($CFG->libdir.'/questionlib.php');
4 require_once($CFG->dirroot.'/mod/quiz/locallib.php'); // We really want to get rid of this
5
6 $id = required_param('id', PARAM_INT); // question id
7 // if no quiz id is specified then a dummy quiz with default options is used
8 $quizid = optional_param('quizid', 0, PARAM_INT);
9 // Test if we are continuing an attempt at a question
10 $continue = optional_param('continue', 0, PARAM_BOOL);
11 // Check for any of the submit buttons
12 $fillcorrect = optional_param('fillcorrect', 0, PARAM_BOOL);
13 $markall = optional_param('markall', 0, PARAM_BOOL);
14 $finishattempt = optional_param('finishattempt', 0, PARAM_BOOL);
15 $back = optional_param('back', 0, PARAM_BOOL);
16 $startagain = optional_param('startagain', 0, PARAM_BOOL);
17 // We are always continuing an attempt if a submit button was pressed with the
18 // exception of the start again button
19 if ($fillcorrect || $markall || $finishattempt || $back) {
20     $continue = true;
21 } else if ($startagain) {
22     $continue = false;
23 }
24
25 require_login();
26
27 if (!isteacherinanycourse()) {
28     error('This page is for teachers only');
29 }
30
31 if (!$continue) {
32     // Start a new attempt; delete the old session
33     unset($SESSION->quizpreview);
34     // Redirect to ourselves but with continue=1; prevents refreshing the page
35     // from restarting an attempt (needed so that random questions don't change)
36     $quizid = $quizid ? '&quizid=' . $quizid : '';
37     redirect($CFG->wwwroot . '/question/preview.php?id=' . $id . $quizid .
38         '&continue=1');
39 }

```

Figure 6.4: Business Logic in VLE (Moodle)

Figure 6.5 proposes a framework for separating presentation logic and business logic. It suggests that presentation logic should be in the client and business logic in the server. The presentation logic is executed before business logic, which will organize the code in both client and server. This will give VLE products the benefits mentioned in the beginning of this chapter.

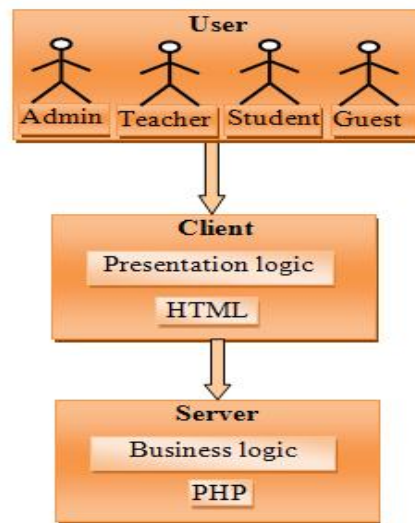


Figure 6.5: Separating Presentation logic and Business logic

6.3.1. Phase II

Phase two, as in Figure 6.1, is an abstraction phase. This phase is the process of reducing the information content of a concept, typically in order to retain only information that is relevant for a particular purpose.

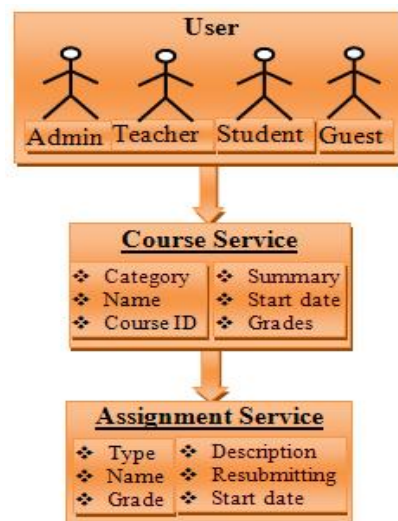


Figure 6.6: VLE after Abstraction

The UML diagram is designed to let developers and researchers view a software system from a different perspective and in varying degrees of abstraction. The

main aim of abstraction in VLE (Moodle) is to reduce and remove all data that is unnecessary from the current Moodle to retain only the information that is relevant for a particular purpose, and the result is shown in Figure 6.6.

6.3.1. Phase III

This phase is the third phase in our model, as in Figure 6.1. This phase is a refinement, which is a fundamental design technique that has often challenged the formal methods. It is the verifiable transformation of an abstract (high-level) formal specification into a concrete (low-level) executable program. Model based refinement is an important technique for ensuring efficiency, effectiveness and correctness in the practical design of complex computer based systems [151].

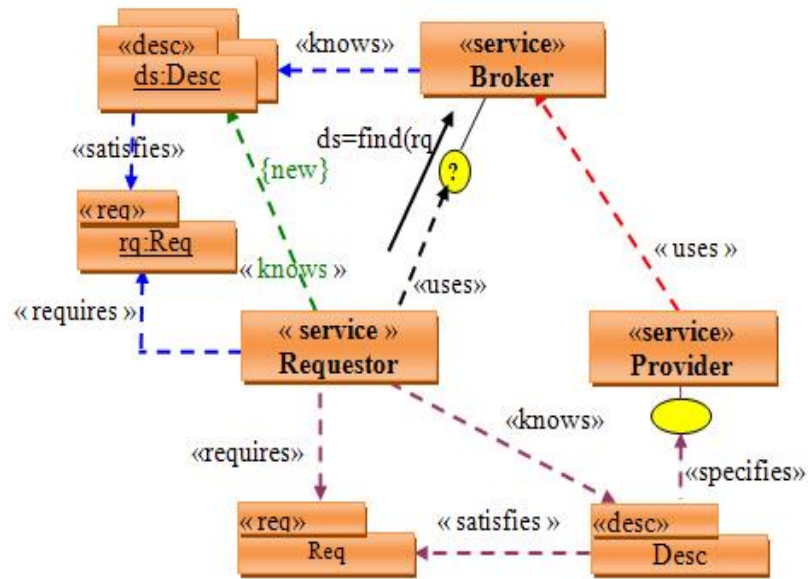


Figure 6.7: Service Oriented Style, Syntax of UML for Web services, and semantics of the Find operation, Service Description and Requester

After abstraction and retaining only data that is relevant for applying Web services, the refinement approach should take place before the final phase. It is important to do this phase in order to check all the data that has been taken from the original model (Moodle).

The roles within the Web services approach are displayed in Figure 6.7, which defines a service as components with stereotype ‘service’. In addition, service requestor and service provider are interpreted as roles of components that implement and utilize the interface of Web services. A service broker is a service that discovers a particular interface that is signed by (?) as short for a stereotype ‘broker’, where the descriptions can be queried by requestors and published by providers [152, 153].

A description of service is a standard UML package signed by ‘desc’, as in Figure 6.7. The above architecture specifies the service and its interface, as indicated by the ‘specifies’ dependencies. Requestors store their requirements for a service in packages assigned by ‘req’. The ‘satisfies’ relation represents the fact that all required properties are guaranteed by a description. Additionally, a ‘knows’ represents the fact that a component (requester or broker) has access to a certain description.

6.3.1. Phase V

This phase is the final phase in our model and it is policy-based request matching; matching users and requests for services that are available in the service registry. As we enable similar services to be registered simultaneously (e.g. course and assignment), distributed among different Moodle servers, matching policies allow the system to assign the most convenient service to the requestor. We use conceptual graphs to model the matching process.

Consider the module application as one part of the VLE tools, which is like an administrator for planning and making an assignment. For ease, we have confined the functionality to making a suitable and realistic assignment, and for this reason, the module software connects to external assignment information systems.

When the browser’s request (admin, teacher, student or quest) has made for a course or an assignment, the system will query all available courses or assignments

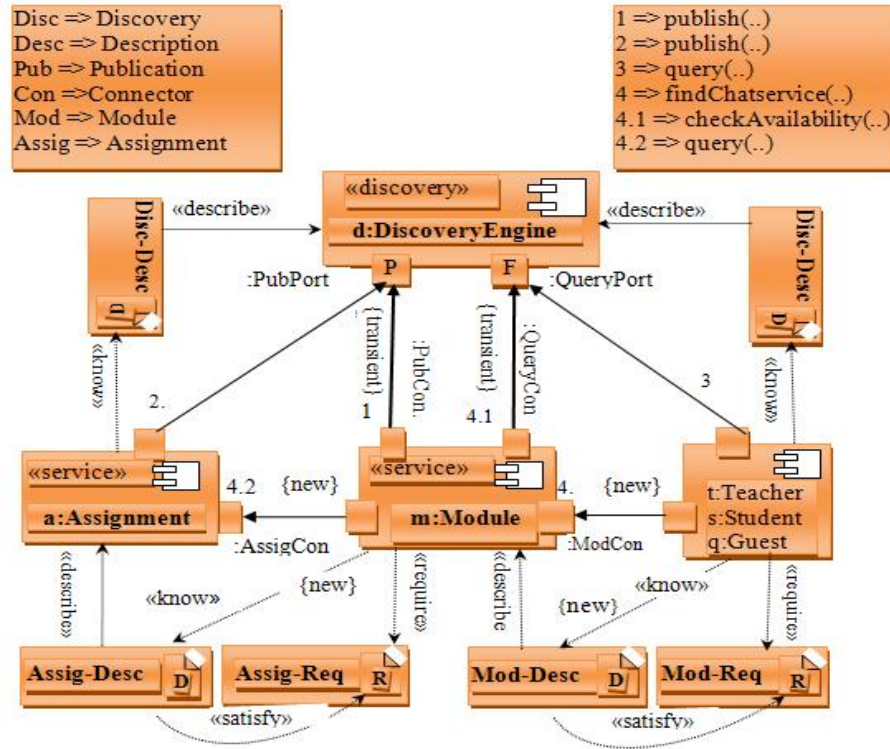


Figure 6.8: Communication Diagram of Web services Based on VLE

in the systems (third party), and after that will choose the optimal assignment according to the browser's request. The requirements guide the user to a distributed and dynamic system where novel components can be brought in by assignment systems at run-time. Therefore, the application that is chosen is a candidate for being realized as a Web service, as modelled with UML for Web services, as in Figure 6.8.

The prototype stage comprises configurations of component instances that are bound to concrete instances at run-time. In Figure 6.8, we use UML collaboration diagrams to model most configurations. We will assign new labels to the affected elements if a set of connections and component instances are changed during the execution. In our example, reconfiguration operations contain the creation of new connections among the components. The communication feature is added to the collaboration diagram by assigning the messages to the connection symbols. We

use numbers to define the ordering of the messages. The specification artefacts, as service requirements and service descriptions, are necessary to allow the dynamic service discovery of Web services [144, 152].

6.4 Matching for Specific Type

The main idea of Web services semantics is to form expressions over the terms supported by ontology. As a formal accuracy is necessary to guarantee a dependable matching of offers and requests, we should use formalism (Graph Transformations), which is in line with the visual representations of ontologies. They unite a formal rigor with a visual appeal and, especially in combination with UML notations, have found a large number of practical applications.

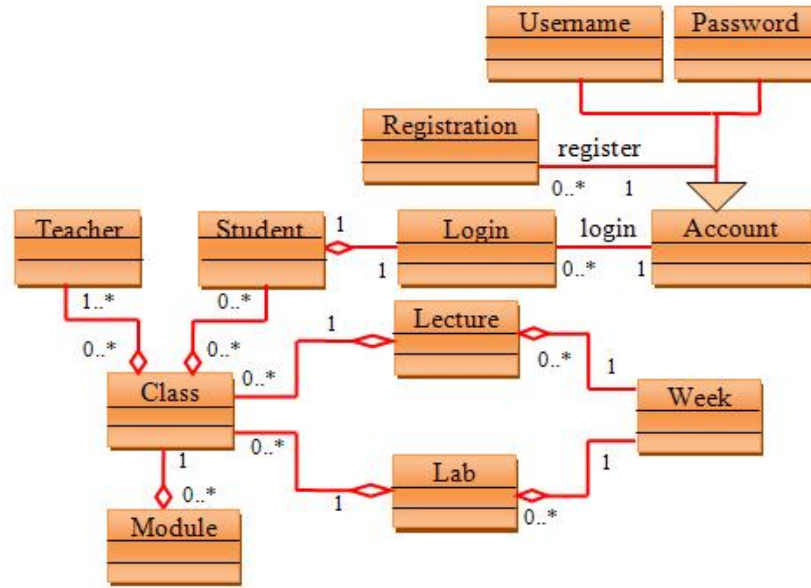


Figure 6.9: Ontology for Registration of Module

The proposed technique is based on ontologies and this mechanism uses graph transformation rules to enable the flexible formulation of new services. Figure 6.9 enhances a simple sample ontology for ‘attend course’. It provides the basic terminology and is depicted as a UML class diagram [154].

The rule in Figure 6.9 displays that student express that his/her service is able to handle for has account in Moodle by registration and get username and password. Figure 6.9 displays the rule that a student who is requesting a service must do so through his/her account, having first registered with a username and password. The rule also affirms that a Web service needs data on a module to be already registered, together with the personal data of the student (username, password and email). These will create a new registration (logged in by username and password) for the specified module and a new entry in the account for this student. With this technique, it is easy to describe all sorts of different services in this area. As real ontologies are far bigger than in our example, many services can be described by constructing this kind of description technique.

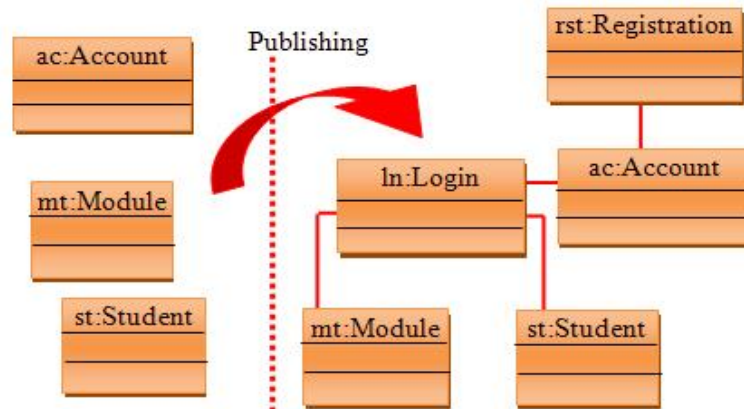


Figure 6.10: Provider Rules

In Figure 6.10, a graph transformation rule consists of two graphs (left and right), which are in our example visualized by using UML object diagrams. Every object that is present in the right hand side is newly created, and every object that is present only in the left hand side, is being deleted. Objects that are present on both sides are unaffected by the rule. If only one object of a type exists, it can remain anonymous, if a distinction between different objects of one type is necessary, then they have to carry an object identifier, separated from their type by a colon. If an even closer resemblance to standard UML concepts is called for, it is also possible

to encode graph transformations in UML Collaboration Diagrams [154].

Graph transformation rules provide both a description of a service and the formulation of a request. The left hand side of a provider position specifies the pre-condition of his service, i.e. the case that has to be present or the information that has to be available for the service to achieve its task. The right hand side describes the post-condition, which distinguishes the situation after the successful execution of the Web service. The left hand side of the rule of a requestor position represents the information he/she is wishing to support to the service, and the right hand side of the rule represents the case he/she wants to achieve by using the service [154].

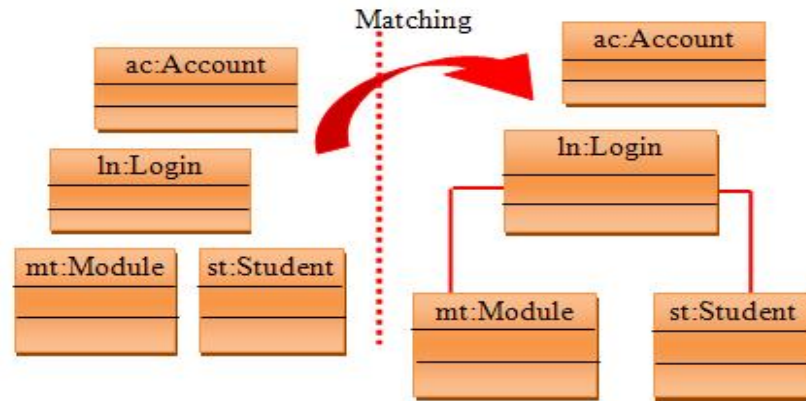


Figure 6.11: Requestor Matching the Specifications

In Figure 6.11 above, we can see such a rule from a requestor's point of view. This rule expresses that the client is able to provide information on a student (an account and his module) and that he/she is looking for a provider; a registration can be constructed based on this information. This means he/she is looking for an institute, and intuitively the provider rule from Figure 6.10 should be a suitable candidate for this request, because in this rule a registration is created [154].

6.5 The Architecture for WSMS with two VLE Platforms (Moodle)

The diagram in Figure 6.12 illustrates the scenario of the WSMS system for two VLE platforms (Moodle), which are AHS and SSiL. This diagram has some components and we will describe them only briefly in this section because we explain them in more detail in Chapter 8.

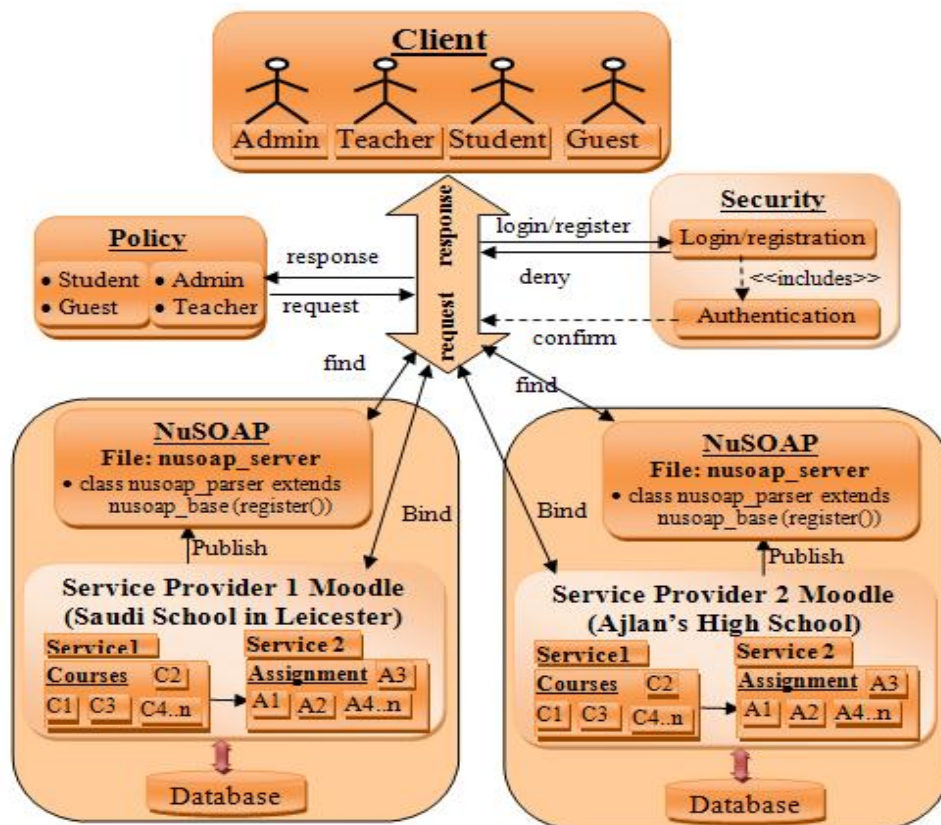


Figure 6.12: The scenario of Web Services for WSMS with two VLE Softwares (Moodle)

Using services in VLE, as shown in Figure 6.12, will help teachers to give their students the opportunity to study with more confidence and to progress in their higher education more easily and more quickly. It also encourages students to share information with their colleagues in the same institution and also with others insti-

tutions across the world. The proposed system also brings easy access to the teacher and student outside official working hours, as well as reducing the administrative burden of the curriculum through the exploitation of the means and tools.

The proposed model in Figure 6.12 has five components, which are client, provider 1 & 2 (Moodle), policy, NuSOAP and security. It offers services for Moodle as a new activity to help students to train themselves anytime and anywhere. This section focuses on these components as follows:

6.5.1. Client

As in the top of Figure 6.12, the proposed model has four main users, which are administrator, teacher, student and guest. These users are described in Figure 6.13 below with their functions:

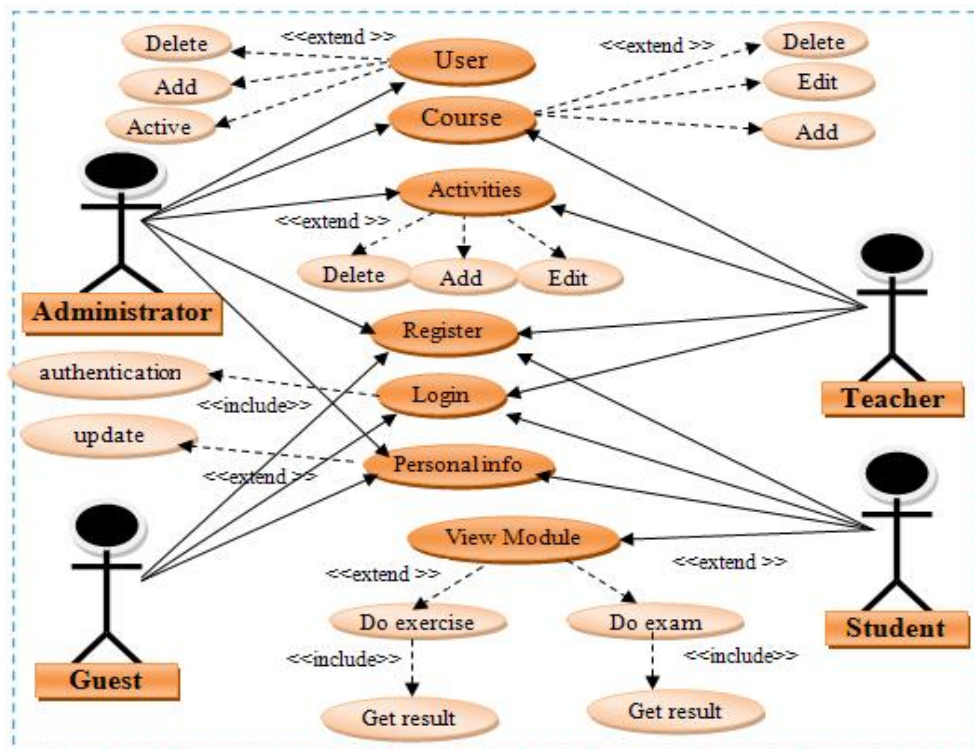


Figure 6.13: Four Users in Proposed Model (administrator, teacher, student and guest)

The first one is administrator who has the ability to use all functions in the system, such as add, delete and update (teacher, student, quest, course, assignment etc). The second user is teacher who can access specific functions and has ability to perform some functions, such as, add, delete and update (course, assignment) to the existing module. The third user is student who can access course, assignment and other activates, as well as update her/his personal information. The fourth user is guest who can view courses and share information with students in the forum, as well as updating her/his personal information.

There are many functions that browsers with different status are able to do and use. Each browser has a definite function that can be performed by the system. Table 6.1 shows all the functions that are provided by our exercise system:

Table 6.1: Features of users in WSMS model

No	Feature	Admin	Lecturer	Student	Guest
1	Manage the Moodle	Y	N	N	N
2	Add, delete and update user	Y	N	N	N
3	Add, delete and update course	Y	Y	N	N
4	Add, delete and update activities (assignment)	Y	Y	N	N
5	Add, delete and update category	Y	N	N	N
6	Add, delete and update personal profile	Y	Y	Y	Y
7	Add, delete and update news	Y	Y	N	N
8	Registration	Y	Y	Y	Y
9	Login	Y	Y	Y	Y
10	View forum	Y	Y	Y	Y
11	View course	Y	Y	Y	Y
12	Add, delete and update grade	Y	Y	N	N
13	View grade	Y	Y	Y	N

6.5.2. Provider 1 & 2 (Moodle)

Services provider is the entity that provides services to other entities. It is the owner of the service and it is responsible for publishing a description of its service to a service registry. It also hosts the service and controls access to it. In our example as

in Figure 6.12, there are two service providers that are hosting in different places (<http://www.saudi-school-le.net/> and <http://www.ajlan-alajlan.com/moodle/>), which are Service Provider 1 Moodle (SSiL) and Service Provider 2 Moodle (AHS). Chapter 8 has more details about these providers and their functions.

6.5.3. Policy

A policy-based technique has become a promising and often-employed solution for managing large-scale systems networks and distributed systems. The most important advantages of using policy-based management are improved flexibility and scalability for the system [152]. The WSMS model has used the environment of ponder language policy, which provides a common means of specifying security policies that map onto various access control implementation mechanisms for firewalls, operating systems, and databases. Ponder can also be used for security management activities such as registration of users or logging and auditing events for dealing with access to critical resources or security violations [152]. Chapter 7 has more information about the Ponder policy and how WSMS applies this policy so that it is suitable for Moodle based on Web services.

6.5.4. NuSOAP

NuSOAP is an important component that enables both schools (AHS and SSiL) to publish and register their services and generate WSDL files. Introducing Web services to Moodle is one of the main objectives of our model as in Figure 6.12. The best way to deliver Web services to Moodle is through the NuSOAP approach. Allowing the NuSOAP package to dynamically generate applications can help a great deal with debugging or adding new service methods. NuSOAP is a good choice for creating and consuming PHP SOAP services for the reasons explained in Chapters 5 and 8.

NuSOAP works as an x-UDDI registry and generates WSDL files. Therefore, it provides information about every service that has registered with it. For example, the `getCourses` function has the details, as in Figure 5.8, when registered with the NuSOAP server. For more information about other functions and their WSDL, please visit http://ajlan-alajlan.com/nusoap/complex_server.php# [155].

6.5.5. Security

Security in WSMS is divided into five levels, which are administrator for WSMS, administrator for schools, teacher, student and guest, as in Figure 5.9 in Chapter 5. Every level has some permission that WSMS has permitted to the users. Our model has strong security as it has authentication, authorization and confidentiality, which makes all schools around the world trust it. Exchanging data over the Internet can be dangerous; it can be fraudulently expropriated or simply lost. Therefore, we are increasing our efforts to combat fraud in the system with a variety of programs.

6.6 Finite State Machine

Finite State Machine (FSM) is used in a range of applications and they can be found in almost all pieces of sequential digital hardware. An FSM goes from one state to another and emits output symbols in response to the input symbols they receive from the environment [156]. Hierons has defined a FSM as [157]:

‘it is model that defines the required behaviour of an implementation, it is important to verify the implementation against the FSM’ [157, pp.1].

An FSM consists of a set of input events, a set of output events, a set of states, an initial state and a set of transitions. It comprises of a data structure used to show actions with a sequence of events. Figure 6.14 shows the basic model of FSM for Web services and contains three states, firstly the initial state is SReq (Service Requestor) and this state is also the final state. The second state is the SReg (Service Registry), which contains all services. The third state is SP (Service Provider), which is the owner of the services.

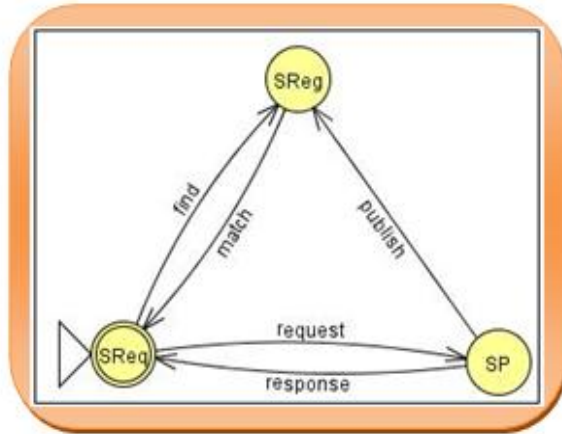


Figure 6.14: Basic model of Finite State Machine for Web Services

The transition between states are as follows:

SP state and SReg state is called ‘publish’.

SReq state and SReg state is called ‘find’.

SReg state and SReq state is called ‘match’.

SReq state and SP state is called ‘request’.

SP state and SReq state is called ‘response’.

This section defines how to describe a FSM using JFLAP. An FSM is made up of input events, output events, a set of states, an initial state and finally a set of transitions. In addition, this section will examine the modelling of the software using FSM, and how users can search for courses and related assignments in a single school or in all available schools, as well as obtaining information about those schools that register with the WSMS system.

6.6.1. Java Formal Languages and Automata Package

Java Formal Languages and Automata Package (JFLAP) is an instructional software used to experiment with grammars automata. The main aspect of JFLAP is that it can experiment with grammars and theoretical machines. JFLAP enables the running and building of user-defined input on pushdown automata, finite automata, and regular grammars. In addition, JFLAP has the ability to construct in steps the proof of the transformation of one form to another form [158].

6.6.2. JFLAP with Design FSM

This section will examine the design of FSM, which has 15 states as in Figure 6.15. The design starts with kind of user (admin, teacher, student and guest). A guest user can only obtain limited results (e.g. name, description of course or assignment) without a login, and if he/she is satisfied with these limited results, he/she can go directly to any school. However, all other users must log in with their username and password as they do in their own schools. These users can view all the details about the courses and related assignments that pertain to them. For example, a

student can only see the grade of his/her course and assignment, just as a teacher can only see the grades of his/her students, but admin can see all the information, as detailed in our examination of policies in Chapter 7. In addition, the design gives the opportunity for these users to see the number of administrators, teachers and students in every school.

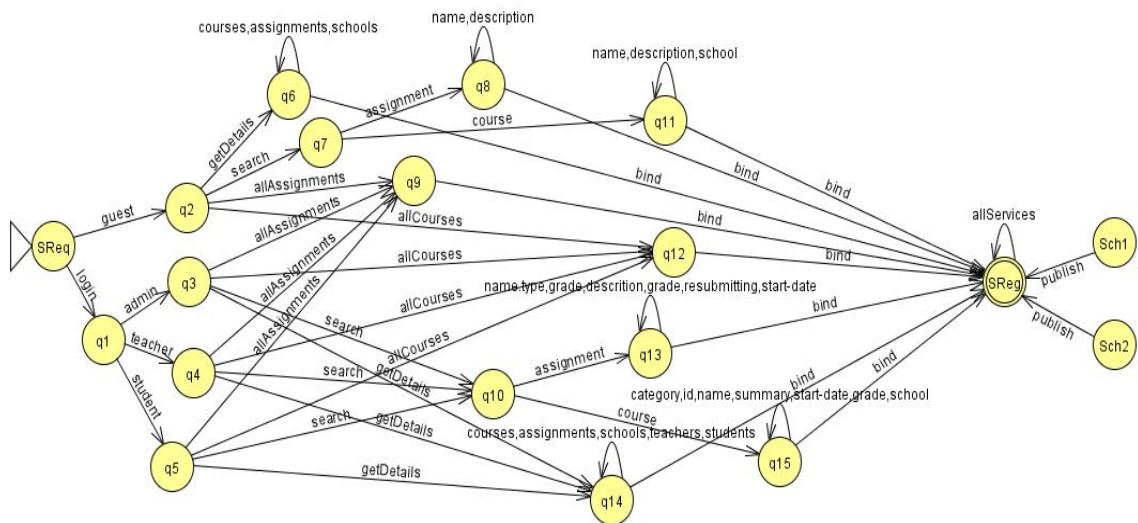


Figure 6.15: The States of WSMS by using Finite State Machine

6.6.3. The Type Test in JFLAP

In JFLAP there are four different types of test. The first type test is step by step, the second is fast test, the third type test is a multiple run test and the fourth type test is a step with closure test. All these type tests show the model is working and covers all the possibilities to test JFLAP. Here, we will use two types, step by step and multiple run, as follows:

6.6.3.1. Step by State Test

The step by state test helps to run the model by using state by state to move to another state if green, eventually to a final state of green. If there are any problems between each state it will be indicated by red. Figure 6.16 shows our example

(loginadminsearchassignmentbind), which starts in the initial state (SReq) and goes to the final state (SReg), passing through q1, q2, q10 and q13.

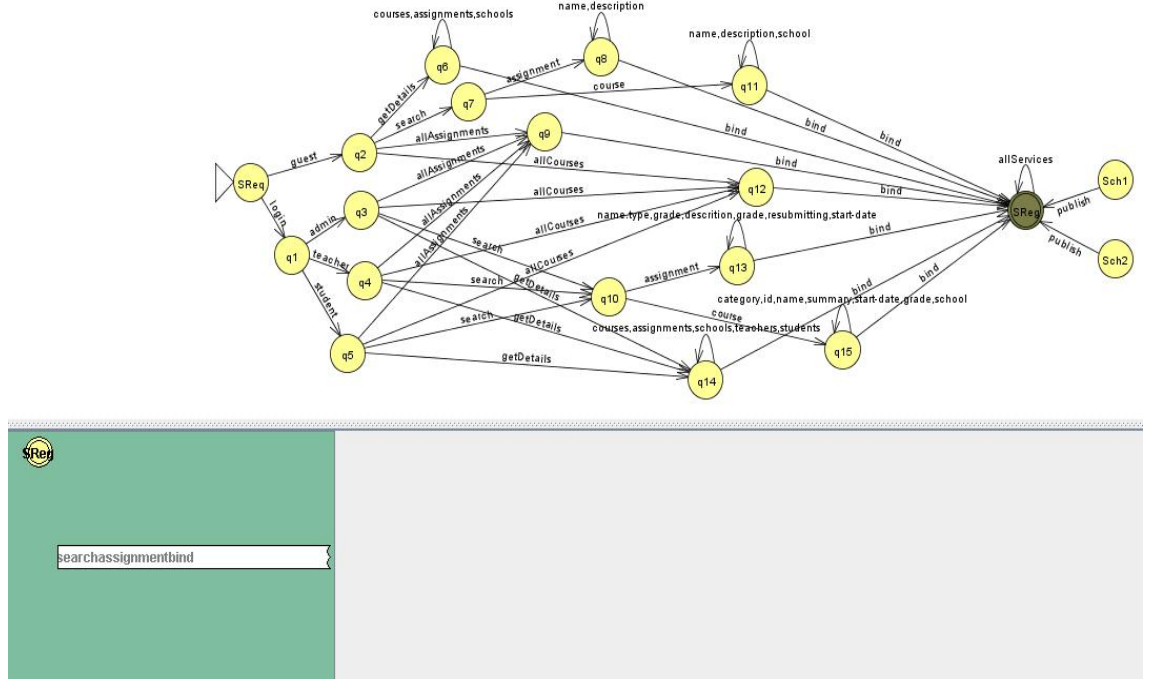


Figure 6.16: WSMS using Step by State in Finite State Machine

6.6.3.2. Multiple Run Test

The multiple run test helps to run all states in the model at the same time and accepts the model if it works correctly and rejects it if there are any problems to be solved, as in Figure 6.17. The traceback of ‘accept’ for WSMS using the multiple run test is displayed in Figure 6.18. This traceback has 11 screens and we have put these screens in one figure, as in Figure 6.18.

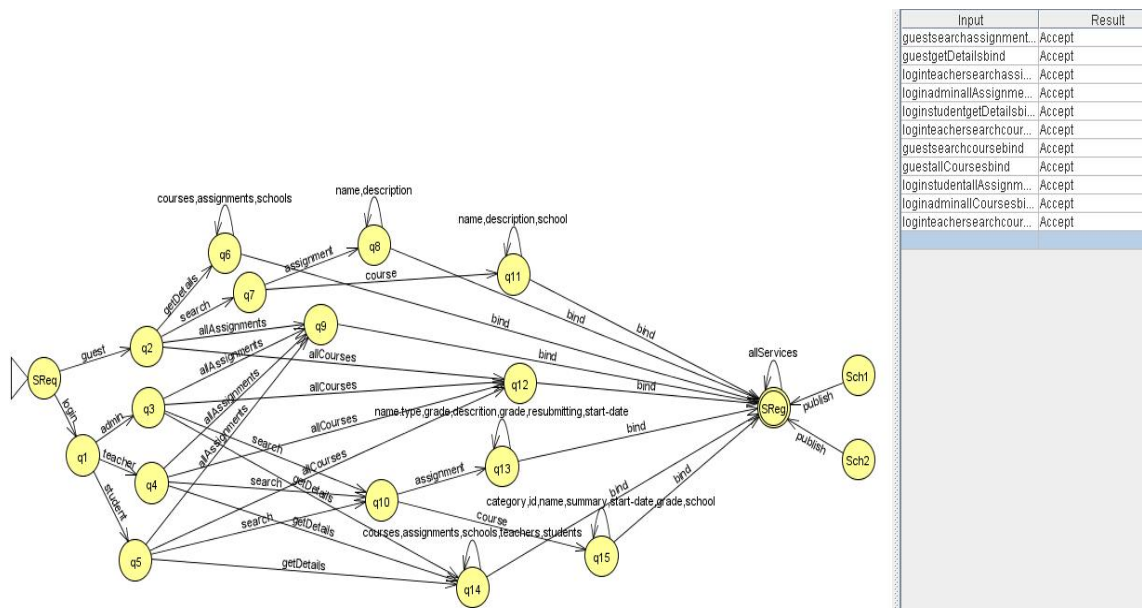


Figure 6.17: WSMS using Multiple Run Test in Finite State Machine

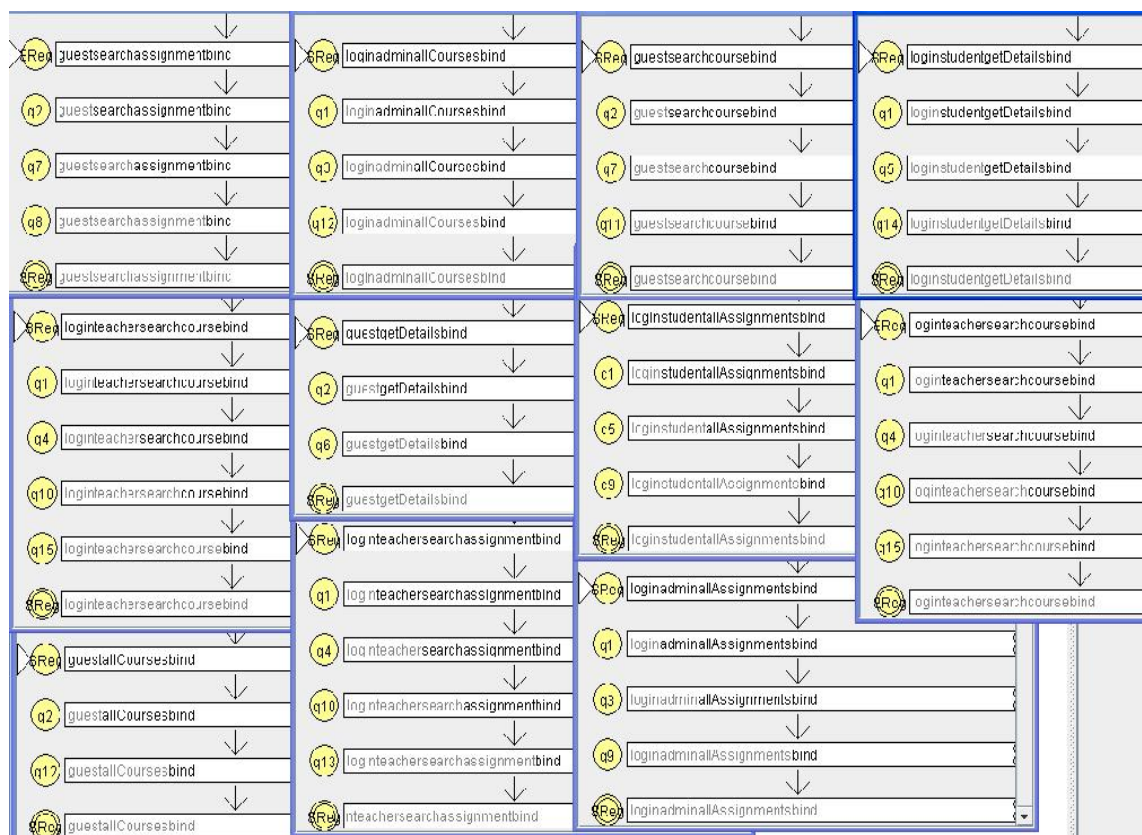


Figure 6.18: Trackback of Accept Test for WSMS Using Multiple Run Test

6.7 Summary

The UML support for SOC sketched in this chapter requires further refinement, in particular, if platform-specific details, e.g. of XML-based Web services, are to be added. This aspect is, however, separated from the purely architectural view of the notation so far, as it is concerned with the contents of service descriptions and queries.

Still, at the architectural level, other types of diagrams are involved, like class diagrams with interfaces to define signatures, and data types of operations and sequence diagrams as alternative presentation of interactions. They are all left out here because of space limitations.

The next chapter will consider the policy that has been used in this research, which is the Ponder policy. This policy will be described in more detail in this coming chapter, and it has two main parts. The first part is access control policies, which has four parts: authorization, information filtering, delegation and refrain. The second part is obligation policies. In addition, this chapter will contain graphical representations of Ponder policies, which also has two aspects. These are domain hierarchy policy and set operation policy. At the end of this chapter, we will explain the tool simulations problem description and AGG simulation.

Chapter 7

Devise A Policy-Based Techniques for Enforce Security Environments of VLE

Objectives

- To provide a comparative study between policy specification languages.
- To provide an overview of the literature on Ponder policies and its types.
- To present a graphical representation of Ponder policy.
- To illustrate problem by using the simulation of tools (AGG).

7.1 Introduction

The use of policy-based techniques has become a promising and often-employed solution for managing large-scale systems networks and distributed systems. The management of large integrated systems, especially trans-national ones, is difficult

because of the wide variety of available policies and the differing types of information that need to be managed. The most important advantages of using policy-based management are improved flexibility and scalability for the system. Flexibility is obtained by separating the policy from the implementation of the managed system, and scalability is improved by uniformly applying the same policy to many sets of devices [29, 159].

Web services are rapidly emerging as a popular standard for sharing data and functionality between loosely-coupled and heterogeneous systems. Therefore, the integration of individual existing Web systems technology allows the provision of advanced and complicated services, such as enabling browsers to use different sorts of resources and services at the same time in an easy procedure. However, the management of a large number of services is not easy; it needs a suitable policy to be defined in order to realize reliable and secure Web services [160, 142].

This chapter proposes a policy verification and analysis method by graph transformations, which support an intuitive way to represent abstract policies in a simple-to-understand style. It acts as an informal language reference for the environment of Ponder policy language. The rest of this chapter is organized as follows: Section 2 presents a brief comparison among policy specification languages. An overview of Ponder policies is described in Section 3. The types of this language are described in Section 3. The representation of Ponder policies by graph rules is described in Section 5. The most important section in this chapter is the simulation of AGG tools with a set of graph rules, and then a policy example can be used to illustrate how to verify and analyse a system with AGG tools.

7.2 A Comparative Study between Policy Specification Languages

A policy is a statement of the objective of the users of computing resources, stating how they desire those resources to be used. Policy specification languages formalize the objectives of the policy programmers into a form that can be interpreted by machines. Therefore, the target of policy specification languages is to give rights to entities (users, programs, etc), permitting some and rejecting others. Every policy has to describe the entities, their attributes and actions or permissions that can be given. Then a single policy binds entities and their attributes to specified actions. Each policy has actions, entities, attributes, and combinations and they can be represented depending on the language used [160, 142].

This section will provide a suitable policy to use it with our approach through a comparative study of the most well-known policy languages. Table 7.1 describes a coarse-grained comparison of some policy specification languages that have been described in this section.

Table 7.1: The comparison between some policy languages

Languages Specification	ASL	ISPS	KAoS	LaSCO	PCIM	Ponder	Rei
Access control	√	√	√	√	√	√	√
Identification	√		√	√		√	
Confidentiality		√					√
Integrity		√					√
Audit				√		√	
Delegation			√			√	
Constraints	√		√			√	
Abstraction	√	√	√	√	√		
Semantics	√		√				√
Reasoning			√			√	
Deployment			√			√	
Editing tools			√				

It can be seen from Table 7.1 that the Ponder, KAoS and Rei languages support a good number of the features required for the security policy specification. Accordingly, this section will describe all policy specification languages that we have described them in our comparison above as in Table 7.1 and these policies are:

7.2.1. Authorisation Specification Language

Authorisation Specification Language (ASL) is a formal logic language. It specifies access control policies and this language includes a form of meta-policies called ‘integrity rules’ to specify application-dependent rules that minimize the set of acceptable access control policies. This language does not work particularly well with large systems because there are no grouping rules in its structures for enabling reusability; a separate rule in this language has to be specified for each action, there is no explicit specification of delegation, and there is no manner of specifying authorisation rules for objects groups.

7.2.2. IPsec Security Policy Specification

IPsec Security Policy Specification (ISPS) applies policy constraints on entities, such as security gateways and router filters with a communication. ISPS specifies confidentiality and integrity rules. It requires some important features such as authentication, audit, delegation etc.

7.2.3. KAoS

KAoS is a collection of services and tools that enable the management, specification, conflict resolution, and enforcement of policies. It uses ontology concepts encoded in OWL to build policies [161]. This language distinguishes between authorization and obligation policies. The applicability of this language is defined by a set of

conditions, and those definitions can have components specifying required history, state and presently undertaken action.

7.2.4. Language for Security Constraints on Objects

Language for Security Constraints on Objects (LaSCO) tries to express constraints on objects. This language is specified as logical expressions and directed graphs. The control aggregation and auditing operations problems are indirectly expressed by this policy. The delegation policy is not supported by these policy syntaxes.

7.2.5. Policy Core Information Model

Policy Core Information Model (PCIM) is an object-oriented information model for distributed systems. The specification of this policy treats the policy-based system as a state machine in which the policies decide the approach in which state transitions take place. It supports an abstract model for defining the structure of policies and relationships among policy objects.

7.2.6. Ponder

This policy is an objected-oriented programming language for using a variety of access control mechanisms for firewalls, operating systems, databases and Java. Section 7.2 above has more details about this policy.

7.2.7. Rei

Rei is a policy framework that integrates support for policy specification, analysis and reasoning. It allows users to express and represent the concepts of rights, prohibitions, obligations, and dispensations.

7.3 Ponder Policies

According to the comparison study above, we have chosen Ponder policy as the suitable language for this project. In this section, we will present the literature on this policy including its detention, benefits and types. Ponder policies for specifying management and security policies evolved out of work on policy management at Imperial College over a period of about 16 years . Ponder is an objected-oriented programming language for using a variety of access control mechanisms for firewalls, operating systems, databases and Java. It supports obligation policies that are event-triggered condition-action rules for policy-based management of networks and distributed systems. In addition, Ponder can be used for security management activities [142, 162].

The key concepts behind Ponder policies are that they contain domains to group the objects to which the policies apply, roles to group policies relating to a position in an organization, relationships to describe interactions between roles, and management structures to describe a configuration of roles and relationships pertaining to an organizational part [137]. Ponder is one of a small number of languages for specifying both management and security policies. It is simple language that uses human managers and its design model is based on domain-based policy management. Domains are hierarchical and are similar to directories. The advantages of the use of domains are [137, 163]:

- They automatically allocate data to sub-domains, which allows them to handle many data sets and thereby to provide scalability.
- They allow new objects to be added/removed from the system without any need to modify policies.

Ponder is a collection of rules that can be used to change system behaviour. It provides positive and negative authorization, information filtering, obligation and

delegation policies. Ponder is a policy specification, which has to be compiled into a programming language for Java and C [137, 161]. Before discussing the policies, we have to introduce some definitions for the terms that are used in Ponder. These terms are subject, target, domain, action, role, relationship, and management structure, and are described in Table 7.2:

Table 7.2: The Syntax and Expression of Ponder Policy

No	Syntax	Expression
	Subject	Student, teacher, institute, and administrator, or system agents.
	Target	education resources or objects such as course, credits, etc.
	Domain	A group of objects which the policies can be applied.
	Action	The activities which subject can carry out. It includes load a new course, enable access, , submit assignment, etc.
	Role	A group of policies which have the same subject, such as the students from the same department.
	Relationship	A group of policies which define the right and duties of roles towards each other, such as the relationship between student and teacher and teacher and administrator etc.
	Management Structure	A group of roles and relationship policies, which defines policy hierarchy of organization structure, such as university and department can be grouped into a management structure.
1	Language Keywords	Bold words (inst- subject-target-action-when).
2	Choices	They are round brackets () separated by
3	Optional elements	They are square brackets [].
4	Repetition	Braces { }.
5	Constraints	It is optional in all types and can be specified to limit the applicability of policies based on time or values of the attributes of the objects to which the policy refers.
6	Elements	They can be specified in any order. The subject and target elements can be optionally. This can be used to check that the objects do support the specified operations or to locate the interface specification.
7	Policy Name	It can be specified as a path, thus identifying the domain into which the policy must be stored.
	in/out keywords	They are used to indicate input and output parameters of the action on which the filter is specified.
	result	It is used to transform the return value of the action.

7.4 Types of Ponder Policy

Ponder is an object-oriented language for specifying management and security policies for distributed system paradigms. It is extensible and flexible, and can cope with the wide range of requirements implied by the current platforms of distributed systems. Here, we are interested more in the basic policy types of this language than in its programming language features. This language has two main types of policies [137, 161]:

7.4.1 Access Control Policies

An Access Control Policy (ACP) is one of the most fundamental and widely used security mechanisms. It authorizes users to perform a set of actions on a set of resources within a system as well as having the ability to permit or deny the use of a particular resource by a particular entity. It can be used to manage many types of resources (physical, logical and digital). This type of policy is concerned with limiting the activity of permitted browsers who have accessed successfully. Ponder policy provides ACP through four policy sub-types, and these are described below [161, 137, 164, 165, 128, 166, 163]:

7.4.1.1. Authorization Policy

The authorization policy is central and the other types are considered as auxiliaries. This type of policy specifies access control for security; it describes a member of the subject domain that can access the group of objects in the target domain in order to protect services and resources from unauthorized access. It has two types:

- A negative authorization policy identifies the actions that subjects are forbidden to undertake on target objects.

- A positive authorization policy identifies the actions that subjects are permitted to undertake on target objects.

Both positive and negative authorization policies must contain subject (except in roles), target and action policy elements. The authorization policy is implemented on the target host by an access control component. The structure of syntax in authorization policy is shown in the Figure below, and the expression of this structure is describes in Table 7.1.

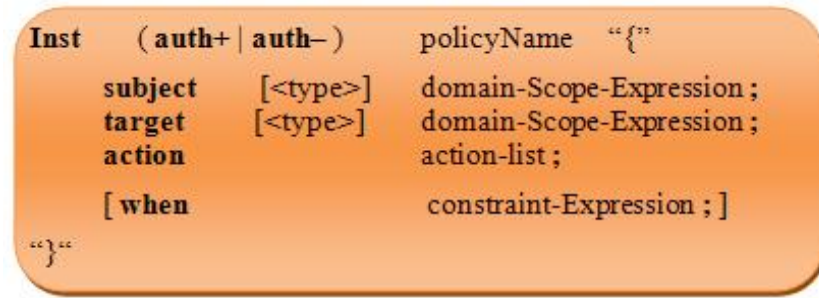


Figure 7.1: Authorisation Policy Syntax

7.4.1.2. Information filtering Policy

Filtering policies transform the data input/output parameters in an action. For instance, a location service may only authorize access to detailed position information, for example a person in an exact area, to browsers within the sub-division. Outside browsers can only determine whether someone is at work or not. Authorization policies (positive) may include filters to transform input or output parameters associated with their actions, based on attributes of the target or subject, on system parameters such as time. Fundamentally, the process has to be achieved and then a choice made on whether to permit the results to be returned to the subject or whether the results need to be transformed. Filtering policies can only be applied to positive authorization actions.

As we can see in Figure 7.2, each action can be associated with a number of

```

actionName { filter }
filter = [if condition] "{ { (in parameterName = expression ; |
                             out parameterName = expression ;
                             |
                             result = expression ; ) } }"

```

Figure 7.2: Filters on Positive Authorisation Actions

filter expressions. Every filter includes an optional condition and if the condition is assessed as being true then the transformations are executed.

7.4.1.3. Delegation Policy

A delegation policy identifies an authorization that allows subject actions to be delegated to others but a user must be strongly controlled by security policies to delegate access rights to another. This kind of policy authorizes subjects to award privileges, which they have to grant to achieve an action on their behalf; it enables cascaded delegation of access rights and it is critical in software.

```

inst deleg+ "(" associated-auth-policy ")" policyName "{ {
    grantee    [<type>] domain-Scope-Expression ;
    [ subject  [<type>] domain-Scope-Expression ; ]
    [ target   [<type>] domain-Scope-Expression ; ]
    [ action   action-list ; ]
    [ when     constraint-Expression ; ]
    [ valid    constraint-Expression ; ]
    "}"

```

Figure 7.3: Delegation Policy Syntax

A delegation policy is related to an authorisation policy that identifies access rights. A delegation policy is not meant to be used as a system of specified user rights by security administrators. As above, delegation has two kinds of policy (positive and negative) where negative delegation policies forbid delegation. Figure 7.3 displays the syntax of a positive delegation policy.

A delegation policy has only one required part and it is the grantee type. This type has three parts: subject, target and action. These parts must be subsets of those in the associated authorisation policy. A positive delegation policy identifies delegation constraints to limit the validity of the delegated access rights. Constraints could be time restrictions, together with duration and validity periods, to indicate the length over which the delegation is valid before it is revoked.

7.4.1.4. Refrain policies

Refrain policies indicate to a subject to refrain from doing something, and are similar to negative authorization policies but are interpreted by the subject. These kinds of policy define the actions that the subjects have to refrain from achieving even though they may actually be permitted to access that target.

```
inst refrain testingRes {  
  subject      s=/test-engineers ;  
  target       /analysts + /developers ;  
  action       discloseTestResults() ;  
  when         s.testing_sequence = "in-progress" ;  
}
```

Figure 7.4: Refrain Policy Syntax

Refrain policies and negative authorisation policies have a similar syntax, but are compulsory for subjects rather than target access controllers. They take action as restraints on the actions that subjects achieve and are executed by those subjects. In addition, they are used for situations where negative authorisation policies are unsuitable because the targets cannot be trusted to enforce the policies. Figure 7.4 shows the syntax of refrain policies, which is the same as negative authorisation policies.

7.4.2 Obligation policies

Obligation policies are the second kind of Ponder policies. These types of policies specify the activities that a subject has to do to a set of target objects, and they define the duties of the policy subject. They are interpreted by an administrator or agent at the subject, and describe the behaviour of the managers of a system when an event happens. Obligation policies define the actions that have to be executed by managers when certain events happen, and they support the ability to respond to changing situations.

These kinds of policies are event-triggered and define the activities that subjects, such as human or automated manager components, have to achieve on objects in the target domain. Events can be an internal and external timer event notified by monitoring service components. Composite events can be specified using event composition operators. The required event specification follows the ‘on’ keyword [163]. The syntax of obligation policies is shown in Figure 7.5.

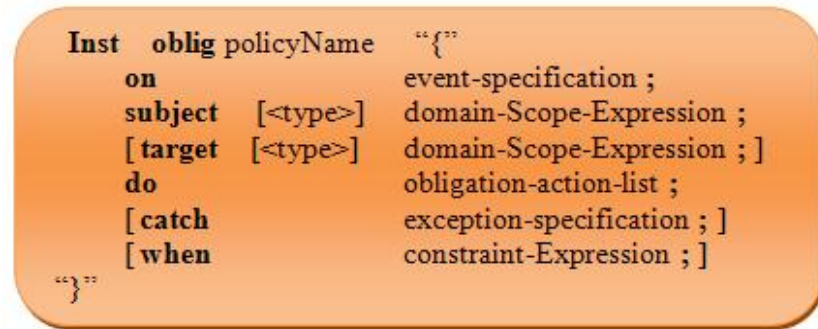


Figure 7.5: Obligation Policy Syntax

As obligation actions, the aim of element is optional (it might be internal to the subject), while authorisation actions always relate to a target object. Actions have to be preceded by a prefix indicating the target set when they are to be invoked on a target. The catch-clause is optional and defines an exemption that is implemented if the actions are unsuccessfully executed for some reason.

7.5 Graph Representation of Ponder Policy

Ponder contains tools for dynamically controlling the behaviour of system components devoid of changing codes and requiring the cooperation of the components being governed. A system should be able to be continuously adjusted to adapt to differences in externally imposed constraints and environmental conditions. A policy-based approach requires a suitable policy representation and the design and development of a policy management framework for controlling a system. Policies will become gradually more important to the real-world implementation of Web services [161, 163]. This part discusses the representation of the basic policy types of Ponder by graph rules.

7.5.1 Domain

Domain is a set of entities that is used to group the system data, and is managed by all the policies. It is arranged in a hierarchical manner, represented by a graph, with nodes for every domain and edges representing the sub-domain relation of an object to a domain. In our graph, the representation has six domains (D1, D2 D3, D4 D5 and D6), and this graph is presented in Figure 7.6.

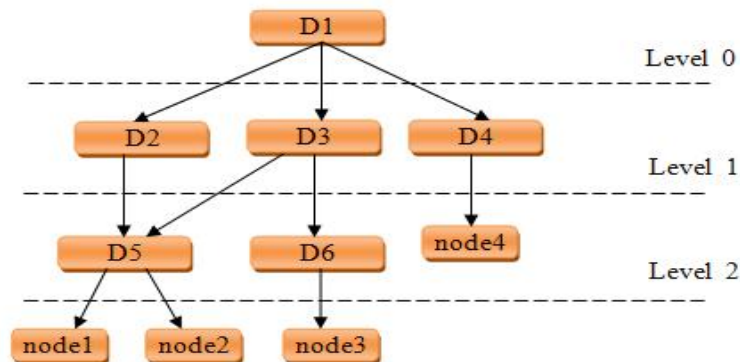


Figure 7.6: The Search paths in Domains

The domain paths as in Figure 7.6 are displayed in Table 7.3. For example, if our goal is Node 1, the search will take the first or third path in Table 7.3

(D1/D2/D5/node1 or D1/D3/D5/Node1).

Table 7.3: The Domain Paths

No	Path	No	Path
1	D1/D2/D5/node1	4	D1/D3/D5/node2
2	D1/D2/D5/node2	5	D1/D3/D6/node3
3	D1/D3/D5/node1	6	D1/D4/node4

7.5.2 Operation

Ponder uses operations of union, difference and intersection to represent domain-ScopeExpressions in a policy. Figure 7.7 represents the operations of Ponder policies for the graph representation of domains in Figure 7.6.

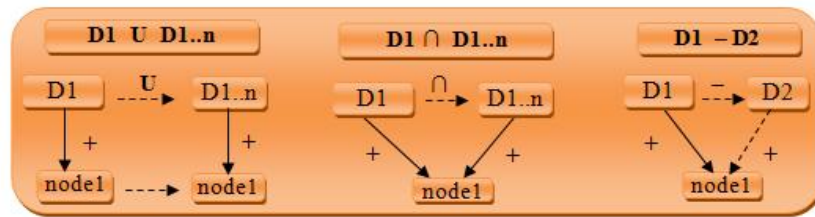


Figure 7.7: Set Operation

- The operation $(D1 \cup D1..n)$, as in Figure 7.7, shows the union operation, with one graph for every component. The symbol '+' means that there exists at least one edge in the path from a domain to a node. When a node satisfies at least one of the graphs, the node satisfies the union.
- The operation $D1 \cap D1n$, as in Figure 7.7, explains the intersection operation. It satisfies the whole graph if a node satisfies the intersection operation.
- The operation $(D1 - D2)$, as in Figure 7.7, explains the difference operation. The dashed edge represents the negative constraint and the solid edge represents the positive constraint such as absence of a path from D2 to node1.

7.6 The Simulation of Tools

Simulation tools are computational models or computer programs that attempt to simulate or model a particular system. They have become a useful part of all areas in the modelling of many natural systems to obtain insights into their operation and to observe their behaviour. Simulation tools are also effective at modelling and analysing the true performance of evolving production systems.

A simulation environment enables developers of established systems to predict behaviour and to apply the tools needed to manage disturbances to an acceptable degree. Thus, simulation tools should intelligently direct us to a smoother and more efficient performance for production systems as well as automatically enabling us to predict the behaviour of most systems and increase production situations [167].

This section describes a simulation tool by using an AGG tool. This tool is a rule based visual language providing an algebraic approach to graph transformation. It will describe a simplified VLE problem with users who are using Web services. In this part, we explain the problem and then use the AGG tool to create a set of rules. Figures 7.9 and 7.10 show the relationship and the scenario for most VLE platforms, and especially for Moodle.

We are using a similar syntax to the Ponder policy (refrain policy type or authorization policy type) as in Figures 7.1 and 7.4 to design our policy as in Figure 7.8, because these types of policies specify the actions that subjects must refrain from performing on the target objects. For instance, if a student submits his/her assignment to teacher with data encryption for delivery, we can use our policy to specify this activity, as in Figure 7.8. The syntax of WSMS contains some definitions for the terms that are used in the WSMS policy. These terms are SUBJECT, TARGET, ACTION, IF, and WHEN. All these terms are explained above in Table 7.2 and the syntax of WSMS policy is:



Figure 7.8: The Syntax of WSMS Policy

7.6.1 The Explanation of Problem

An understanding of the relationships between the nodes is important for building the scenario of the WSMS system in VLE products. Therefore, before we start to analyse the problem, we should know the position of every node in the VLE (Moodle), as in Figure 7.9.

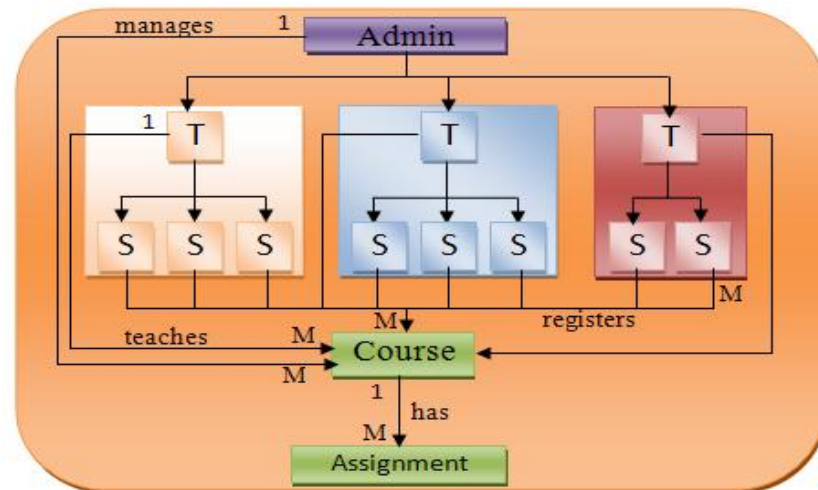


Figure 7.9: The Relationship between users and course in VLE (Moodle)

Admin is the supervisor of any VLE system and especially Moodle. He/she can invoke any target such as add, update, delete, etc. Teachers can access and control their courses including activities (assignments), and can communicate with students. Students can access their courses and related activities including assignments, as in Figure 7.9. Table 6.1 in Chapter 6 has more details about the targets of users.

7.6.2 AGG Simulation

AGG tools are a rule based visual language providing an algebraic approach to graph transformation. They aim to the specification and prototypical implementation of applications with complex graph-structured data. These tools have some characteristics that are [165, 168]:

1. Complex data structures are modelled as graphs by a type graph.
2. The system's behaviour is identified by graph rules using (if-then).
3. Application of several rules sequentially shows an application scenario.

Figure 7.10 shows the type graph of WSMS by using AGG tools. This graph describes the scenario of the problem, which is with some type nodes: Users node (Admin, Teacher, Student and Guest) Course node, Assignment node, AssignmentG node, Account node, Payment node, Login node and Request node.

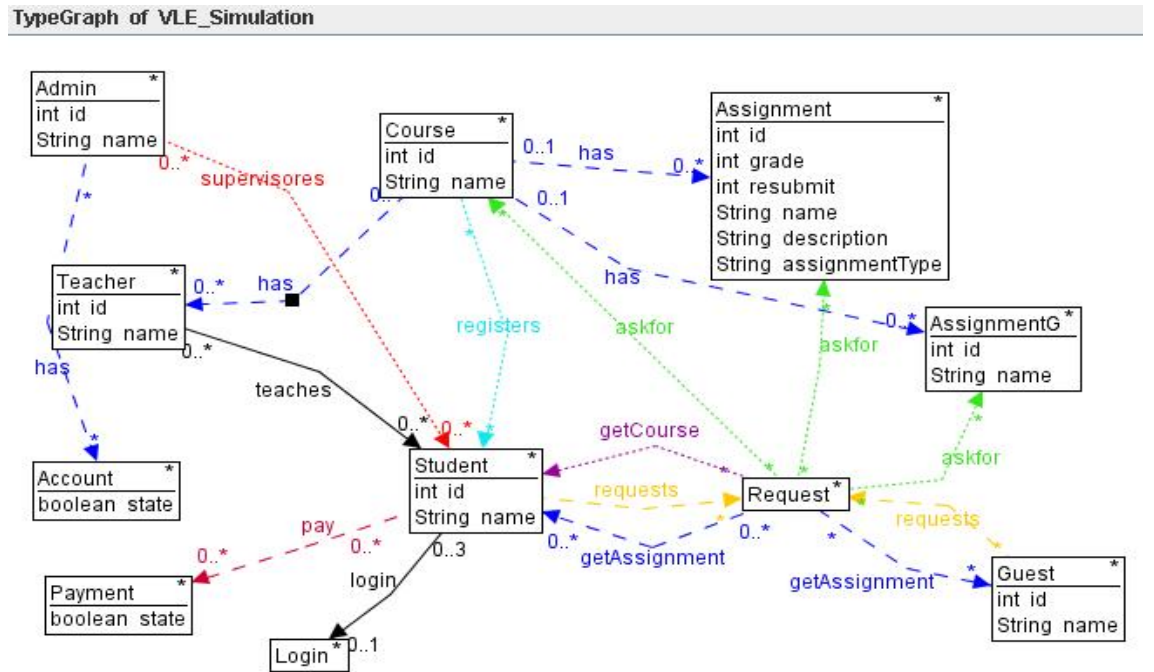


Figure 7.10: The Type Graph of WSMS

The most important nodes in our type graph for WSMS (Moodle) are the course, student and request nodes. We describe this type with the rules and their policies below.

7.6.3 The Rules and their Policies of the WSMS System

In this part, we set up some rules according to the scenario in the type graph of WSMS as in Figure 7.10. We also build a policy for each rule depending on our policy syntax, as in Figure 7.8. The tool that has been used to set up these rules is an AGG tool as described in the above section. The abbreviations (t, s, c, ac and a) in our policy indicate student, course, account and assignment respectively. The left cell in each rule is the negative application condition, the middle cell is the left side, and the right cell is the right side.

Here, we will mention only five rules; the other rules are in Appendix D. The first rule gives the student node with condition ($x = \text{register}$) the right for a student to enrol with that course node. The second rule allows all the student nodes to log in by giving them three attempts ($x = 3 \text{ time}$) to log in with the account node. The third rule enables the student node to access the assignment node with condition ($x = \text{s.registers}(c), \text{s.has}(c), \text{s.request}(a) \text{ and } \text{s.askfor}(a)$) through the binding service. The fourth rule enables the student node to access the course node with condition ($x = \text{s.registers}(c), \text{s.has}(c), \text{s.request}(c) \text{ and } \text{s.askfor}(c)$) through the binding service. The fifth rule enables the teacher node to manage the course node with condition ($x = \text{t.has}(c), \text{s.has}(s)$). These rules and their policies are as follows:

7.6.3.1 The First Rule

The first rule permits the student node to enrol with the course node as in Figure 7.11. The condition of this rule is that the student must be registered with the course, satisfying the condition if ($\text{s.registered} = \text{true}$); so only the student node

with `student = register` satisfies this rule and binds the service. The left cell in Figure 7.11 is the negative application condition of this rule, and again the middle cell is the left side and the right cell is the right side.

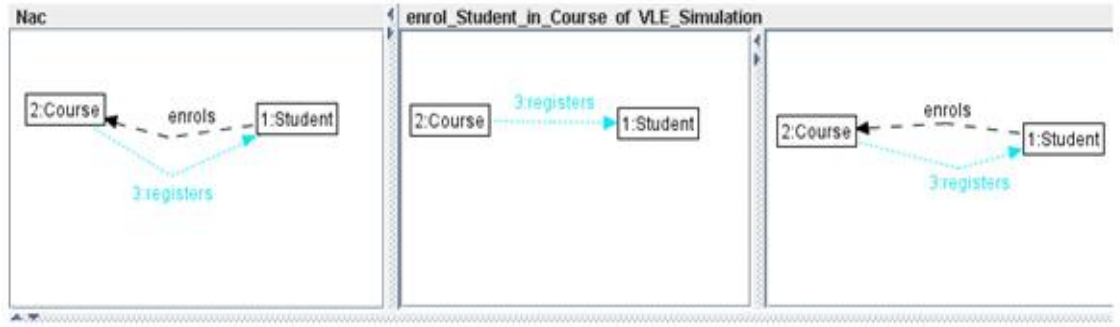


Figure 7.11: Rule 1-The enrolment of a student

Rule: enrol_student.

Description: This rule says that a student cannot enrol on a course unless that student is already registered with the teacher for that course.

Parameters: Course (id, name) and student (id, name).

The policy of rule 1 is described in Figure 7.14. This policy allows a student to enrol on a course node only if that student meets the condition: if `student = s.registers(c)`; then the policy enables the student to bind the service.



Figure 7.12: The Policy of Rule 1

7.6.3.2 The Second Rule

Figure 7.13 describes rule 2, which permits the student node to log in only up to and including three times, thus allowing only three attempts. The condition is $s.\text{login}(3)$, so only the student node with $\text{student} = \text{login}$ 3 times satisfies this rule.

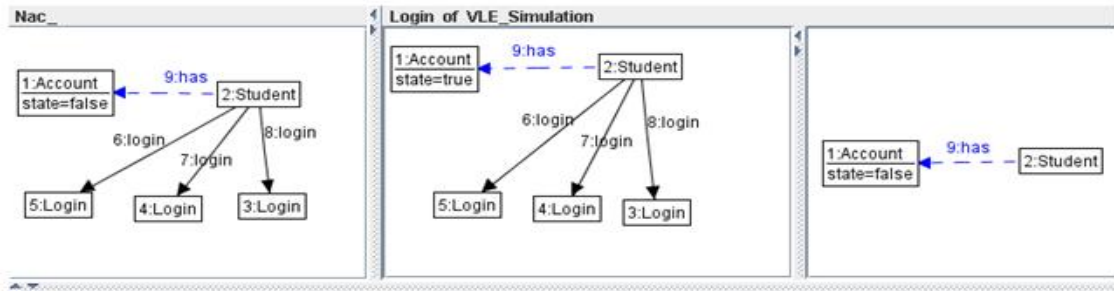


Figure 7.13: Rule 2 - the login attempts of a student

Rule: login.

Description: This rule enables students to try three times to log in into their account, and then the account will lock.

Parameters: Account (Boolean), student (id, name) and login.

The policy of rule 2 is described in Figure 7.14. This policy allows a student to log in into an account node only if the student meets the condition: if $\text{student} = s.\text{login} = \text{true}$; then the policy enables the student to bind the service and access the account.

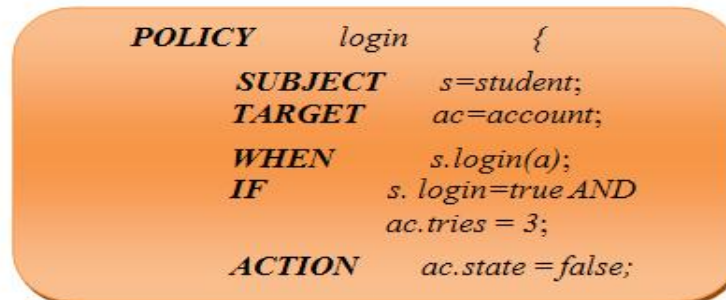


Figure 7.14: The Policy of Rule 2

7.6.3.3 The Third Rule

The third rule shows how a student can access the assignment service by binding the request service for a student, as in Figure 7.15. The condition of this rule is that the student must `s.register(c)`, `s.request(a)`, `c.has(a)` and `s.askfor(a)`. If a student satisfies all these conditions, the rule will allow the binding service (AssignmentG node), as in Figure 7.15.

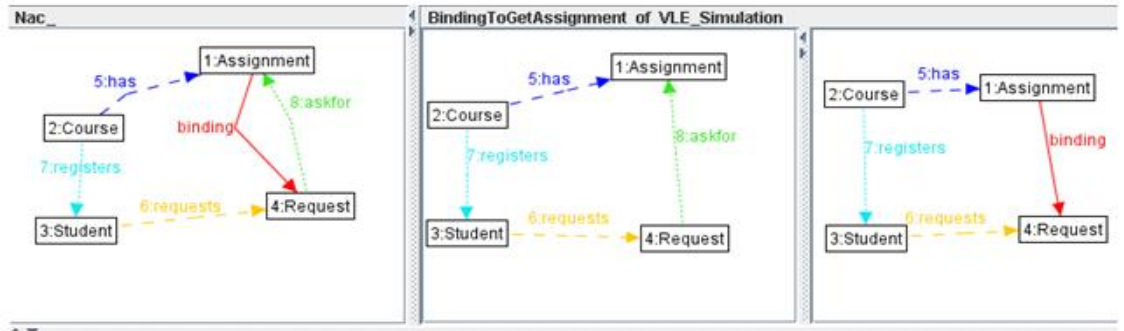


Figure 7.15: Rule 3 - the binding to access the assignment service

Rule: Binding to access assignment.

Description: This rule enables students to access the assignment service only if they have registered with the course that has this assignment, requested and asked for this assignment.

Parameters: Course (id, name), student (id, name), assignment (id, name, grade, description, assignmenType and resubmit).

The policy of rule 3 is described in Figure 7.14. This policy allows a student into the assignment node only if the student meets all the conditions: if student = (s.register(c), s.request(a), c.has(a) and s.askfor(a)); then the policy enables the student to bind the service.

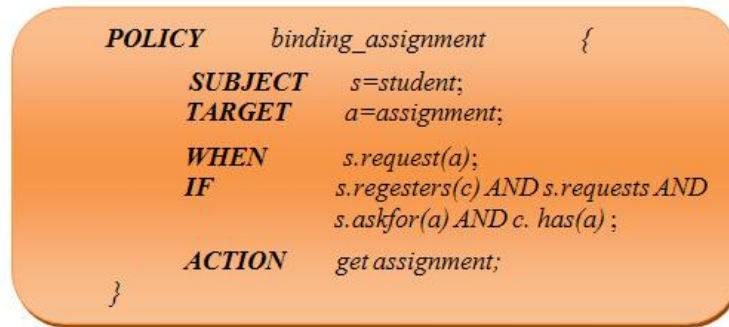


Figure 7.16: The Policy of Rule 3

7.6.3.4 The Fourth Rule

The fourth rule is similar to rule 3 in that it shows how a student can access a course node by binding the service, as in Figure 7.17. The condition of this rule is that the student must s.register(c), s.request(c) and s.askfor(c). If the student satisfies all these conditions, the rule will allow the binding service (Course node).

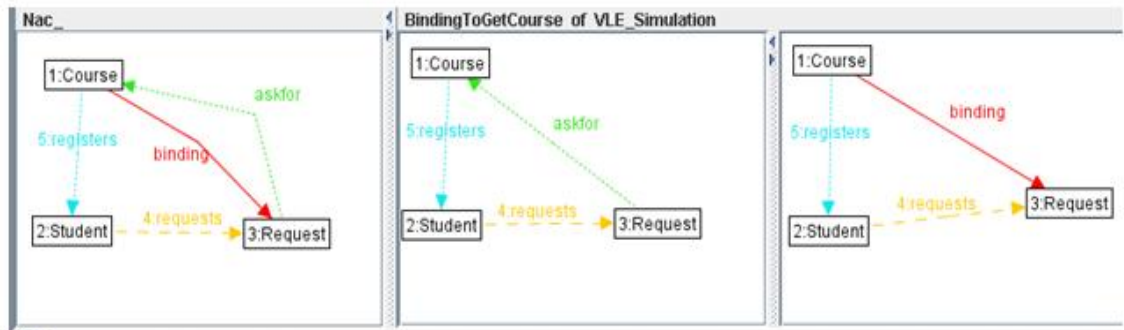


Figure 7.17: Rule 4 - the binding to access the course service

Rule: Binding to access course.

Description: This rule enables students to access a course if they have already registered on the course, requested and asked for this course.

Parameters: Course (id, name), student (id, name), request.

The policy of rule 4 is similar to the policy of rule 3, and it is described in Figure 7.18. This policy allows a student to access the course node only if the student

meets all the conditions: if student = (s.register(c), s.request(a) and s.askfor(a)); then the policy enables the student to bind the service.

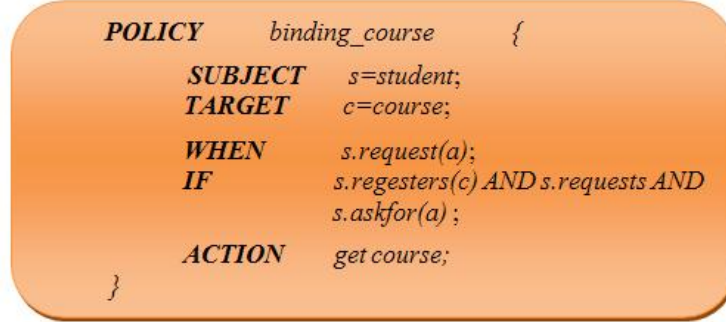


Figure 7.18: The Policy of Rule 4

7.6.3.5 The Fifth Rule

The fifth rule describes how a teacher can manage a course node, as in Figure 7.19. The condition of this rule is that the teacher must t.has(c), and t.teaches(s). If the teacher satisfies all these conditions, the rule will allow the teacher to manage the course node.

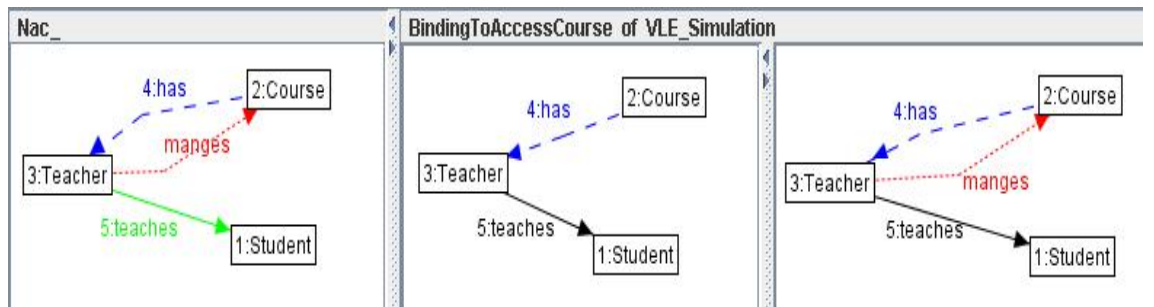


Figure 7.19: Rule 5- Teacher manages course

Rule: managing course.

Description: This rule enables teachers to manage a course only if they responsible for this course and also have students enrolled (not an empty course).

Parameters: CTeacher (id, name), course (id, name), student (id, name), request.

```

POLICY      manage_course      {
    SUBJECT  t=teacher;
    TARGET   c=course;
    WHEN     t.mange(c);
    IF       t.has(c) AND t.teaches(s);
    ACTION   manage course;
}

```

Figure 7.20: The Policy of Rule 5

By using AGG tools we can execute our rules and the result of this execution is displayed in Figure 7.21.

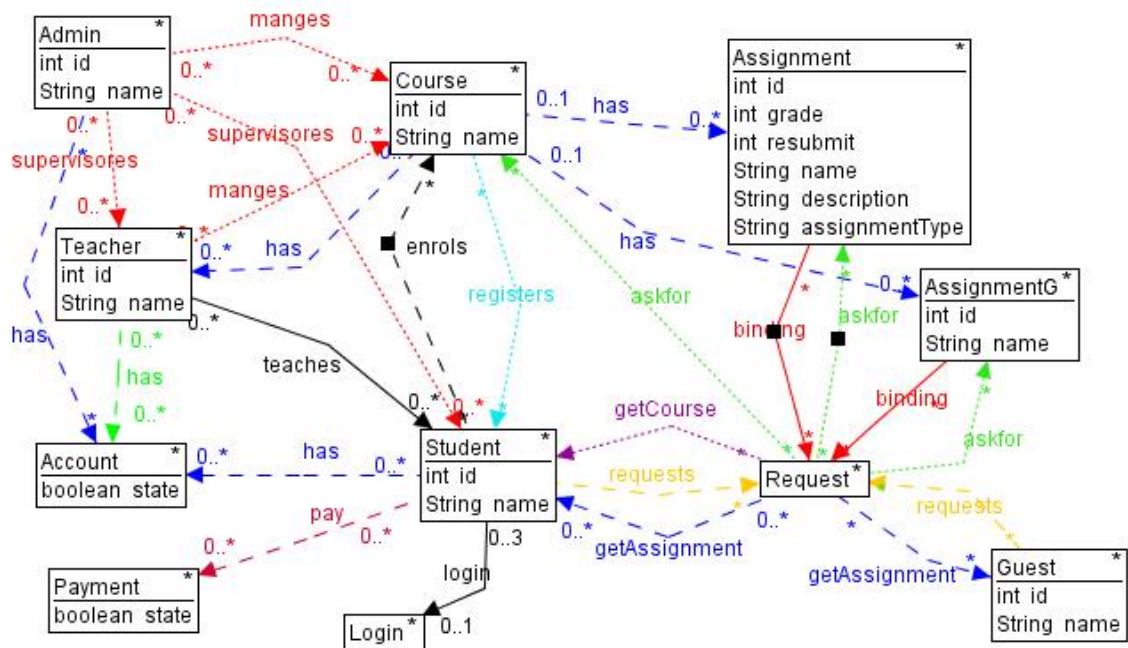


Figure 7.21: The Result of Execution Type Graph of WSMS System

7.7 Summary

This chapter has presented a comparison study between policy specification languages. According to this comparison of the most well-known policy languages, we have chosen Ponder as suitable policy specification languages to use it in this project. Ponder is a language for specifying policies for the management and security of distributed systems. Ponder includes authorization, filter, refrain and delegation policies for specifying access control, and obligation policies to specify management actions. Ponder thus provides a uniform means of specifying policy, relating to a wide range of management applications, networks, storage, systems applications, and service management.

This chapter has proposed a Ponder policy verification and analysis method by graph transformations. It acts as an informal language reference for the environment of the Ponder policy language. In addition, we have provided an overview of Ponder, including the types of this language and a brief comparison with some other policy languages. We have discussed in more detail the representation of Ponder policies by graph rules, and we have used the simulation AGG tools with a set of graph rules, and then a policy example was used to show how to verify and analyse a system with those AGG tools.

In the next chapter, we will give an overview on some techniques that support developers to create SOAP with PHP language. It will describe how to extend PHP Moodle to support Web services by using the NuSOAP technique. Moreover, it will illustrate that this use of Web services is a straightforward evolution of Web programming models, and it will demonstrate how PHP can be used as a fast and easy development tool for creating them. In the end of this chapter, we will provide an overview of the most commonly used classes and methods in the WSMS system through the implementation of Web services in order to support PHP Moodle by using NuSOAP technique.

Chapter 8

The Extension of Web Services to Support PHP Moodle by Using NuSOAP Technique

Objectives

- To provide an overview of PHP Web Services Implementation Toolkits.
- To focus on the NuSOAP package (SOAP Toolkit for PHP).
- To introduce the Web services approach to VLE.
- To explain the implementation of Web services to support PHP Moodle.

8.1 Introduction

PHP is one of the easier programming languages; it is Object Oriented (OO), and shows developers how to take full advantage of these new features. The PHP structure is fixed and very clear; it relieves developers from all the complexities of memory

management and of dealing with the existing versions of C language. Most of the rules and structure of PHP have been taken from the C, Perl and Java languages for the creation of a smooth and easy-to-use language without any loss of power [169].

These days, PHP has become the preferred choice for developing SOC, especially for Web services. Servers provide applications as services with XML data that are written in PHP functions in order to implement these services and register these functions with SOAP implementation; this automatically makes the PHP functions remotely accessible. The outcome of this is that service owners can focus on improving and developing the services that they want to implement it in the server. Thus, the connection between the client and the server, and the parsing of XML messages, will all be automated [138]. Today, there is no standard API for developing Web service applications, but there are some techniques that can help developers to develop Web services. At the moment, the most popular techniques are NuSOAP, ezSOAP, PEAR::SOAP, PHP SOAP and NuSphere [59].

This chapter will give an overview on some techniques that support developers to create SOAP with PHP language. It will describe how to extend PHP Moodle to support Web services by using the NuSOAP technique. In addition, it will illustrate that this use of Web services is a straightforward evolution of Web programming models, and it will demonstrate how PHP can be used as a fast and easy development tool for creating them. Finally, this chapter will provide an overview of the most commonly used classes and methods in the WSMS system through the implementation of Web services in order to support PHP Moodle by using NuSOAP technique.

8.2 PHP Web Service Implementations

Web services enable enterprises to collaborate in distributed computing with external business partners. They are an evolution of the standards and protocols used to create the Web as we know it today, and are a group of emerging and established communication protocols that consist of XML, SOAP, UDDI and WSDL over HTTP [170, 171]. One of the greatest advantages of Web services is that they allow a number of applications to be integrated more quickly, easily and cheaply than ever before. They are expressed as WSDL, which is an XML-based language. A service may be published and discovered using UDDI, while SOAP allows vendor-neutral communication between applications over HTTP [171].

These days, there is no standard API of PHP language for developing Web services (SOAP) applications but there are some techniques that can help developers to develop Web services. The most popular ones at the moment are NuSOAP, ezSOAP, PEAR::SOAP, PHP SOAP and NuSphere. This section will give a brief description of these techniques, and the next section will focus on NuSOAP because our research will apply this approach to Moodle (PHP)[59].

8.2.1. NuSOAP

NuSOAP is a group of PHP classes that allows developers to create and consume SOAP Web services. It is a good choice for creating and/or consuming PHP SOAP services. The next section (8.3) will give more details about this package and will explain why it has been chosen for this study [172].

8.2.2. PEAR::SOAP

This package is distributed under the PHP license. The SOAP implementation delivered by PEAR in this package offers a basic, yet very useful way of devel-

oping SOAP clients and servers with PHP. It is similar to NuSOAP because it is based on SOAPx4. On the other hand, this implementation has been developed a little further, and is probably a little harder to understand. The installation of PEAR::SOAP does not require any additional extensions, as the library will make use of the overload extension if it is loaded. The latest version of PEAR::SOAP is 0.7 and it requires PHP 4.1 or later [59, 173].

8.2.3. ezSOAP

This kind of technique is free software, and is distributed under the GNU General Public License (GPL). GPL obliges developers and researchers to supply the complete source code of their package, even if they distribute it commercially. Therefore, if anyone plans to use ezSOAP for commercial software, they need to purchase the ezPublish professional license. With the PHP web-publishing suite ezPublish (from ezSystems), comes a generic SOAP implementation called ezSOAP. Just like the other SOAP implementations, ezSOAP provides a rather simple way of creating and consuming SOAP services [59].

8.2.4. PHP SOAP Extension

PHP SOAP Extension is one of the most popular PHP implementations of SOAP 1.1 and 1.2, developed by the PHP Group. It supports WSDL, is easy to use, and two statements are enough to call a SOAP service; www.xmethods.net offers a good demonstration of SOAP service. The installation of this package is included in the PHP 5.0.4 package for Windows, distributed by Zend Technologies [174].

8.2.5. NuSphere

This package is an Internet Application Platform (IAP) based on open-source components. It supports an integrated foundation that enables businesses to deploy

reliable, cost-effective, enterprise-class applications across Linux, UNIX and Windows operating systems. One of the many advantages of NuSphere is its integrated software suite, which pairs the reliability and cost effectiveness of PHP, Apache, Perl and open-source databases in order to build business-critical web applications and Web services. NuSphere was named the ‘Best Developer Tool’ at LinuxWorld in the autumn of 2001; it is based in Bedford, Mass., and the company’s commercial software services include training, technical support and consultation [170].

8.3 NuSOAP (SOAP Toolkit for PHP)

This package is distributed under the Lesser General Public License (LGPL). NuSOAP is a group of PHP classes that allows developers to create SOAP. It can generate WSDL files and also to consume them for use in serialization. Both rpc/encoded and document/literal services are supported. On the other hand, it must be noted that NuSOAP does not provide coverage of SOAP 1.1 and WSDL 1.1, which are as complete as other implementations such as Apache Axis and .NET.

NuSOAP and SOAP implementation work with each other in an easy manner by automatically handling difficulties, and by supporting a fair amount of access to the flexibility and nuance functions underneath. NuSOAP is provided by NuSphere and Dietrich Ayala, and is a rewrite of SOAPx4. It is a group of classes written in the PHP language and there is no need for PHP extensions. This package enables developers to build Web services based on SOAP, WSDL and HTTP. A numbers of current classes of SOAP allows researchers and developers to deal with all information inside NuSOAP [138, 175, 59, 172].

The reasons for choosing this package over other packages are [59, 138, 172, 175]:

1. It provides more power and control for PHP programmers who use SOAP. Sustained improvement of document service provides WSDL/XSD compliance,

and this makes NuSOAP suitable for the formation of powerful Web services for services, containing well-liked profitable services (Amazon and Google).

2. NuSOAP does not require any special PHP extensions and comes under LGPL. It is a rewrite of the SOAPx4 library and comes in a 140K PHP file.
3. It delivers a complete SOAP implementation for PHP, without relying on any extra PHP extensions, which makes it easy to use.
4. NuSOAP is easy to install for non-root users with tested servers. It has decent compatibility with both PHP4 and 5.

There are a number of diverse implementations of SOAP under PHP language. It is an uneven landscape where new SOAP implementations appear, and old ones disappear. The NuSOAP implementation is the most commonly used and appears to be the most fully developed and actively maintained, and it shows every sign of continuing to be a robust and well-liked solution. It is not complete in terms of features, including full documentation, but is still being developed. Nevertheless it is extremely simple and viable to make use of SOAP implementation [59].

8.3.1 The Installation of NuSOAP Package

The developers can obtain the NuSOAP Package from SourceForge or the CVS tree of the NuSOAP project on SourceForge. A browser-based CVS interface is provided. Once developers have downloaded a copy of nusoap.php and they simply need to place it in their code tree so that they can include it from their PHP code. Some researchers and developers put it in a separate lib directory, but in our examples in this research we placed it in the same directory as the sample code itself.

When running PHP, one needs a PC to download NuSOAP from the Internet, available free from the developer's site. The extension of the installation file is zipped and should be opened before use in order to obtain a documentation folder;

the nusoap.php file has the real PHP classes that researchers and developers need. In this study, we have used the nusoap.php file with both schools

8.3.2 The Architecture of NuSOAP Package

The architecture of the NuSOAP package has nine classes, as in Figure 8.1, and all these classes are in one PHP file called nusoap.php. The main class is nusoap_base, which has 28 methods, as in Appendix A. The authors of this package is Dietrich Ayala (dietrich@ganx4.com) and Scott Nichol (snichol@users.sourceforge.net), and the version that we are working with is \$Id: nusoap.php,v 1.94 2005/08/04 01:27:42 snichol Exp \$. Appendix A has all the details about the NuSOAP package, including the classes and all their functions.

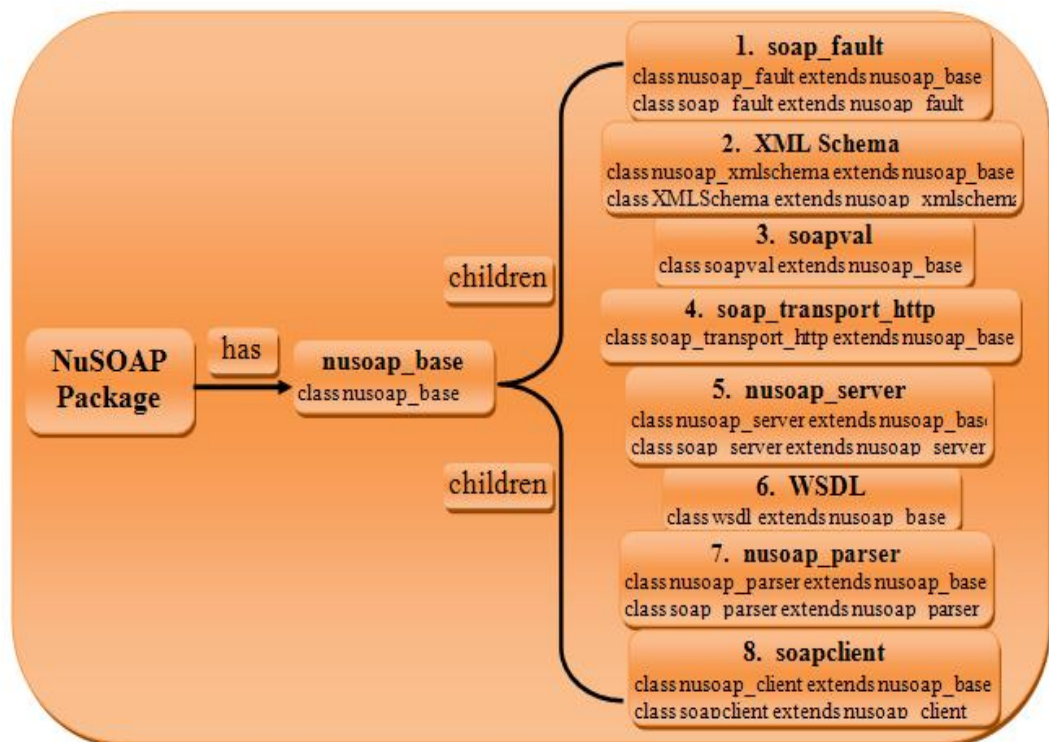


Figure 8.1: The Architecture of NuSOAP Package

8.4 Introducing the Web Services Approach to Meet VLE (Moodle)

Recently, the use of IT in VLEs in the higher education institutions has increased, and so VLEs should become fully embedded into e-learning and teaching practice instead the traditional approaches to VLEs. However, this movement within VLE systems is not yet fully realised, and the most important challenges to this target are retaining learners within education, and encouraging a wider group of learners to engage in higher education. One of the approaches that has been taken to meet these challenges is to use the Web services approach to encourage closer working and cooperation between higher education institutions. The most widely accepted and successful type of Web service is XML documents. This type of service has two essential requirements: it communicates via Internet HTTP protocols, and it sends and receives data formatted by XML messages [134, 135].

The proposed approach is aimed at extending the current architecture of Web services to meet the technical requirements of Moodle, as in Figure 6.12 in Chapter 6. The aim of this approach is to explain how we introduce the Web services to meet the VLEs (Moodle). The most important advantage of this approach is that it allows users at different institutions in different countries to share material and work together by connecting courses together, even though they are hosted on different VLEs. Figure 6.12 has major components that are necessary if one is to use services in Moodle 1 and 2, and we explain them in more detail here:

8.4.1. Service Provider

Service provider is the entity that provides services to other entities. It is the owner of the service and it is responsible for publishing a description of its service to a service registry. It also hosts the service and controls access to it. In our example,

there are two service providers that are hosted in different places (www.saudi-school-le.net/ and www.ajlan-alajlan.com/moodle/), which are Service Provider 1 (SSiL) and Service Provider 2 (AHS), as in Figure 6.12.

8.4.1.1 Service Provider 1 Moodle (SSiL)

In our model (WSMS), the SSiL (Moodle) will be Service Provider 1. This provider will publish its services, all courses and related assignments in service its registry. When the service requestor searches for his/her specific order, the WSMS will reduce the service selection by using the attributes of both consumer and service, and offer the optimal one that meet requestor's needs.

8.4.1.2 Service Provider 2 Moodle (AHS)

As in SSiL (Moodle), so will it happen in the second service provider, which is AHS (Moodle). It will also publish its services, all courses and related assignments in a service registry to be available to a service requestor. Also, here, the WSMS will offer the optimal one that meet requestor's needs.

8.4.2. Service Registry

The service registry plays an important part in our example; it is a NuSOAP package and works as an x-UDDI registry. We will consider it as a central store that make possible service discovery by service requestors. This component provides a searchable repository of service descriptions where service providers publish their services, and service requesters find services and obtain binding data for these services. It attracts many services from providers and offers them, enabling requestors to find and use a variety of services that meet their needs. The main goals of service registry are to support a friendly and simply-accessible mechanism for the publishing and discovery of services, to build and maintain trust in both requestor and

provider communities, and to match as many clients as possible to providers that offer services of interest to the clients. [169].

8.4.2.1 Web Services Matching & Selection System

The WSMS system begins with the discovery of a set of services by a potential requestor, and a correspondence is then established between the objectives of the requestor and the capability of the service. Here, we do not need to discover all available services matching a set of requirements and specifications; the system just needs to locate one that is good enough in terms of features and quality. WSMS offers a new extension to the structural design of Web services, which keeps data on the Web services, frequently exploring them to collect data about their functional and non-functional requirements. Chapter 5 and 6 has more details about WSMS.

8.4.2.2 Policy

The most significant advantages of using policy-based management are improved flexibility and scalability for the system, especially large-scale systems. The WSMS system uses the environment of Ponder language policy, which provides a common means of specifying security policies that map onto a variety of access control implementation mechanisms for firewalls, operating systems and databases [176, 137]. Chapter 7 has more information about the Ponder policy and how WSMS applies this policy to be suitable for Moodle based on Web services.

8.4.3. Service Requestor

A service requestor is called a service consumer, and it is a software component in search of the service to invoke across the Web. It finds the most suitable service by discovering the set of available services that meets some pre-defined criteria. A service requestor is the sender of a Web service message or the software program

requesting a specific Web service. In WSMS, the service requestor is admin, teacher, student and guest [33].

8.4.4. Course & Assignment Details

As shown in Figure 6.12, Moodle in both SSiL and AHS has many details, and therefore the service providers in both will publish just the necessary data; they published the details of their courses and assignments services, as in Table 8.1. This data is available for a requestor as a service in the service registry (x-UDDI (NuSOAP)).

Table 8.1: The details of courses and assignments

Service	Attribute	Description
Course	Category	Administrator may have set up several course categories. This choice will affect where course is displayed on the course listing and may make it easier for students to find a course.
	Name	The full name of the course is displayed at the top of the screen and in the course listings.
	Course ID	The ID number of a course is only used when matching this course against external systems, as in our approach, which depends on this ID to match with other VLE systems
	Summary	Writing text about courses also has the ability to include URL addresses and some HTML tags in text.
	Start date	Teachers specify the starting time of the course (depending on their own time zone).
	Grades	Many of the activities allow grades to be set. By default, the results of all grades within the course can be seen on the Grades page, available from the main course page.
Assignment	Type	There are four types of assignment as in Figure 4.11.
	Name	The full name of the assignment as in Figure 4.10.
	Description	Writing text about assignment also has the ability to include URL addresses and some HTML tags in text.
	Grade	Individual posts can be rated using a scale, based on the theory of separate and connected knowing.
	Allow resubmitting	By default, students cannot resubmit assignments once the teacher has graded them. If a teacher turns this option on, then students will be allowed to resubmit assignments after they have been graded.

Table 8.1 shows the details of the courses and assignments that are the names of the service attributes, together with a description of these attributes. Table 8.2 displays where this information is in the database, both in Moodle 1 and 2. It gives

more details about the information: names of tables, fields and types.

Table 8.2: The location of details of courses and assignments (table, field and type)

No	Table	Field	Type
1	mdl_course	id	bigint(10)
2	mdl_course	category	bigint(10)
3	mdl_course	fullname	varchar(254)
4	mdl_course	summary	text
5	mdl_course	category_description	bigint(10)
1	mdl_assignment	id	bigint(10)
2	mdl_assignment	course	bigint(10)
3	mdl_assignment	name	varchar(254)
4	mdl_assignment	description	text
5	mdl_assignment	assignmenttype	varchar(50)
6	mdl_assignment	resubmit	tinyint(2)
7	mdl_assignment	grade	bigint(10)
1	mdl_course_categories	id	bigint(10)
2	mdl_course_categories	name	varchar(254)
3	mdl_course_categories	description	text
1	mdl_assignment_submissions	id	bigint(10)
2	mdl_assignment_submissions	assignment	bigint(10)
3	mdl_assignment_submissions	userid	bigint(10)

8.5 The Implementation of Web Services to Support Moodle by Using NuSOAP Approach

The PHP language has been become a popular option for programmers using Web services, especially in VLE systems, and therefore, the introduction of Web services to Moodle is one of the main objectives of our study. Recently, the developers of Moodle have been facilitating the discussion about the development of Web services, and as a consequence of this, we have concluded that the implementation of Web services to support PHP Moodle needs one of the techniques mentioned in Section 2 of this chapter. This technique is NuSOAP (SOAP Toolkit for PHP). It is one of the most popular ones at the moment for the reasons mentioned in Section 3 of this chapter. Allowing NuSOAP to dynamically generate applications can help a great

deal with debugging or adding new service methods, and thus NuSOAP is a good choice for creating and/or consuming PHP SOAP services.

The server will support applications with data in a standardized approach that uses PHP functions to implement these services. Then functions will register with NuSOAP implementation, which will mechanically make PHP functions remotely accessible. The XML messages will be generated automatically by the NuSOAP package implementation. The outcome will let developers to concentrate on improving and developing the services that they would like to implement in the server. The communication between the server and client, and the parsing of XML messages, will all be considered automatically. The XML structure will become a standard multi-dimensional PHP array and it can easily iterate to retrieve all values [59][2].

Central to the implementation is AHS, which has many files in (www.ajlan-alajlan.com/moodle/) as in Table 8.3 and some of these files are PHP files. One of them is the Cascading Style Sheets (CSS), which is a stylesheet language used to describe the presentation of a document written in a markup language.

Table 8.3: The Files in WSMS System and all Schools

No	File	Description
1	complex_server.php	This file is important and every school must have this file in its server. It has complex type, registration services and all functions with select query.
2	search.php	This file is 'control the search' and has search courses, embed the search form, hide login, show results if courses available, get school URL and get counts for user types functions.
3	nusoap.php	This is the heart of WSMS, and Appendices A & B have more details.
4	ajlan.css	To control the frame of pages. It describes the presentation of a document written in a markup language.
5	assignment_grades.php	To control the grades between users.
6	assignments.php	To get assignment details.
7	login.php	To enable users to log in.
8	login_verify.php	To check users.
9	course.php	To get course details.
10	logout.php	To log out and destroy session.
11	students.php	To get student details.
12	top_links.php	It has all links in WSMS to avoid the duplication in every file.
13	searchform.php	It is the form of search as in Figure 8.12.
14	course_details.php	It has all courses details in all schools that register in WSMS

Enabling Web services for more VLE system applications allows educators, students and guests at different institutions in different countries to share material and work together by connecting individual courses together, although hosted on different VLEs. In our example, there are two kinds of VLEs, which both use the same VLE system: Moodle. However, and in addition to this, our example can deal with many VLEs as services by adding `complex_server.php` (the VLE name) and `nusoap.php` to the school server.

This section will focus on the above files and will describe in more detail the target of each class and function. In addition, Appendix A has a full code of the NuSOAP library.

8.5.1 Using WSDL

The NuSOAP file should be placed in the `system/libraries` folder (we assume we already have there the `nusoap` class (`nusoap.php`)). We have to explicitly specify which types are being used by the functions, and we describe the functional approach for registering the types. Below is the WSDL configuration for a NuSOAP server:

```
require_once('nusoap.php');

$server = new soap_server();
$server->configureWSDL('serverwsdl', 'urn:serverwsdl');
$server->wsdl->schemaTargetNamespace = 'urn:serverwsdl';
```

Figure 8.2: The WSDL configuration for a NuSOAP server

8.5.2 Complex Types for WSDL

The service registry (NuSOAP) in WSMS not only receives basic data types and returns single values, but it can create complex data types to allow different structures to be passed to and from the server. The `addComplexType` method registers structures and arrays that contain a structure `Course`, `Assignment`, `Assignment-`

GradesInput, AssignmentGrade, Student, UserDataInput and UserCounts. Here, a more complex data type is created to return more detailed courses and assignments:

8.5.2.1. The ComplexType for Course and ArrayOfCourses

The following code is related to AHS and registers a structure Course, which contains five members (an int ID, a string fullname, a string summary, a string category and a string category_description) and an array of Course structures. It is created to return more details around the courses:

```
// Add Course & ArrayOfCourses Complex Data Types
//*****
$server->wsdl->addComplexType(
    'Course',
    'complexType',
    'struct',
    'all',
    '',
    array(
        'id' => array('name' => 'id', 'type' => 'xsd:int'),
        'fullname' => array('name' => 'fullname', 'type' => 'xsd:string'),
        'summary' => array('name' => 'summary', 'type' => 'xsd:string'),
        'category' => array('name' => 'category', 'type' => 'xsd:string'),
        'category_description'=>array('name'=>'category_description','type'=>'xsd:string')
    )
);
$server->wsdl->addComplexType(
    'ArrayOfCourses',
    'complexType',
    'array',
    '',
    'SOAP-ENC:Array',
    array(),
    array(
        array('ref'=>'SOAP-ENC:arrayType','wsdl:arrayType'=>'tns:Course[]')
    ),
    'tns:Course'
);
```

Figure 8.3: The Code ComplexType for Course and ArrayOfCourses

8.5.2.2. The ComplexType for Assignment and ArrayOfAssignments

The following code registers a structure Assignment, which contains six members (an int ID, a char name, a string description, a string assignmenttype, a string resubmit and a string grade) and an ArrayOfAssignment structure. It is created to return more detailed assignments:

```
// Add Assignment & ArrayOfAssignments Complex Data Types
//*****
$server->wsdl->addComplexType(
    'Assignment',
    'complexType',
    'struct',
    'all',
    '',
    array(
        'id' => array('name' => 'id', 'type' => 'xsd:int'),
        'name' => array('name' => 'name', 'type' => 'xsd:string'),
        'description' => array('name' => 'description', 'type' => 'xsd:string'),
        'assignmenttype' => array('name' => 'assignmenttype', 'type' => 'xsd:string'),
        'resubmit' => array('name' => 'resubmit', 'type' => 'xsd:int'),
        'grade' => array('name' => 'grade', 'type' => 'xsd:int'),
        'school' => array('name' => 'school', 'type' => 'xsd:string')
    )
);

$server->wsdl->addComplexType(
    'ArrayOfAssignments',
    'complexType',
    'array',
    '',
    'SOAP-ENC:Array',
    array(),
    array(
        array('ref'=>'SOAP-ENC:arrayType','wsdl:arrayType'=>'tns:Assignment[]')
    ),
    'tns:Assignment'
);
```

Figure 8.4: The Code ComplexType for Assignment and ArrayOfAssignments

8.5.3 Registering Functions for WSDL

Functions will register with the NuSOAP and we need to specify the incoming parameters/types and the return type. The following functions, getCourses, searchCourses, courseDetails and getAssignments, take a string parameter title and return a course and assignment structure. Here, we will just mention getCourses and getAssignments services.

```
// Register Functions
//*****
$server->register('getCourses', // method name
    array(),
    array('return'=>'tns:ArrayOfCourses'),
    'serverwsdl', // namespace
    'serverwsdl#getCourses', // soapaction
    'rpc', // style
    'encoded', // use
    'Get all current courses' // documentation
);
```

Figure 8.5: The Registration of getCourses Method


```
$server->register('getAssignments',      // method name
    array('courseid' => 'xsd:int'),      // input parameters
    array('return'=>'tns:ArrayOfAssignments'),
    'urn:serverwsdl',                  // namespace
    'urn:serverwsdl#getAssignments',    // soapaction
    'rpc',                              // style
    'encoded',                          // use
    'Get relevant assignments based on a course ID' // documentation
);
```

Figure 8.6: The Registration of getAssignment Method

8.5.4 The WSDL File

When the WSDL file specification is complete, it is possible to examine it by passing the WSDL parameter to the PHP script. The script will return the generated WSDL, which can be used to generate proxy classes. Figure 8.7 shows the functions that have been registered as Web services in WSMS and also gives more details about the specifications of the getCourse function. Appendix B has more information about the WSDL code for this example and also it is available on the following link: http://ajlan-alajlan.com/nusoap/complex_server.php?wsdl



Figure 8.7: The functions that have registered as Web services in WSMS system

8.5.5 Implementing Functions

By using the WSDL specification, we can return SOAP structures as regular PHP associative arrays. When the server handles a SOAP request, it serializes and unserializes the PHP arrays using the WSDL description. The following function returns a course with its members:

```
function getCourses() {
    mysql_connect('localhost','ajlanal','gLv9f9ow3Vjxu3');
    mysql_select_db('ajlanal_mdle1');
    $query = "SELECT cs.id, cs.fullname, cs.summary, cg.name as category,
               cg.description as category_description"
            ." FROM mdl_course cs"
            ." Left Outer Join mdl_course_categories cg on cg.id = cs.category"
            ." Where cg.id > 0";
    mysql_query("SET NAMES 'UTF8'"); //So important for Arabic Text
    $result = mysql_query($query) or die(mysql_error());
    $returnThis = array();
    while($row = mysql_fetch_array($result)){
        $tempCourseObject['id'] = $row['id'];
        $tempCourseObject['fullname'] = $row['fullname'];
        $tempCourseObject['summary'] = $row['summary'];
        $tempCourseObject['category'] = $row['category'];
        $tempCourseObject['category_description'] = $row['category_description'];
        array_push($returnThis, $tempCourseObject);
    }
    return $returnThis;
}
```

Figure 8.8: Courses connects with database (SQL)

8.5.6 Database Services

Database services are important and they are responsible for making connections with the database. Each school has special database services that connects with the school's database and obtains the data that we specify in every service, as in Figure 8.9. WSMS has many database services such as getCourses, searchCourses, courseDetails, getAssignments, getAssignmentGrades and getStudents.

Moodle uses Structured Query Language (SQL), which lets users access data in relational database management systems, for example Microsoft SQL Server, Oracle, Sybase, Informix, and others. It allows users to describe the data that the requestor wants to see, and it defines and manipulates the data in that database [177].

```
// All Functions
//*****
//GET USER DETAILS IF EXISTS
function getUserData($login) {
    mysql_connect('localhost','ajlanal','gLv9f9ow3Vjxu3');
    mysql_select_db('ajlanal_mdlel');
    $username = $login["username"];
    $password = $login["password"];
    $query = " SELECT u.id, u.firstname, u.lastname, u.email, r.name as type "
        ." FROM mdl_user u "
        ." INNER JOIN mdl_role_assignments ra ON ra.userid = u.id "
        ." INNER JOIN mdl_role r ON r.id = ra.roleid "
        ." WHERE u.username = '$username' and u.password = '$password' "
        ." LIMIT 1";
    mysql_query("SET NAMES 'UTF8'"); //So important for Arabic Text
    $result = mysql_query($query) or die(mysql_error());
    $num_rows = mysql_num_rows($result);
    if ($num_rows > 0)
    {
        $row = mysql_fetch_array($result);
        $user['id'] = $row['id'];
        $user['firstname'] = $row['firstname'];
        $user['lastname'] = $row['lastname'];
        $user['email'] = $row['email'];
        $user['usertype'] = $row['type'];
        $user['school'] = getSchoolName();
    }
    else
        unset($user);
    return $user;
}
```

Figure 8.9: Data Source and How It Connects with Database

```
$query = " SELECT CONCAT(us.firstname, ' ', us.lastname) as student, s.grade, "
        ." CONCAT(ut.firstname, ' ', ut.lastname) as teacher "
        ." FROM mdl_assignment_submissions s "
        ." //To get student name "
        ." INNER JOIN mdl_user us on us.id = s.userid "
        ." //To get teacher name "
        ." INNER JOIN mdl_assignment a on a.id = s.assignment "
        ." INNER JOIN mdl_course c ON c.id = a.course AND c.category <> 0 "
        ." INNER JOIN mdl_course_categories cg ON c.category = cg.id "
        ." INNER JOIN mdl_context con ON con.instanceid = c.id AND con.contextlevel= 50 "
        ." INNER JOIN mdl_role_assignments ra ON ra.contextid = con.id "
        ." INNER JOIN mdl_role r ON ra.roleid = r.id AND r.name = 'Teacher' "
        ." INNER JOIN mdl_user ut on ut.id = ra.userid "
        ." Where s.assignment = '$assignmentid' "
        ." AND ( '$usertype' = 'Administrator' "
        ."       OR ('$usertype' = 'Teacher' AND ut.id = '$userid') "
        ."       OR (us.id = '$userid') "
        ." ) ";
```

Figure 8.10: The Select Query of searchCourse Service in WSMS

Moodle has a very large database, which has more than 198 tables in Version 1.9. Each database service has select query and this select depends on some of the Moodle tables and the relationships between them. For example, the searchCourses service has a long query and six tables, as in Figure 8.10, and the relationships between these tables are displayed in Figure 8.11.

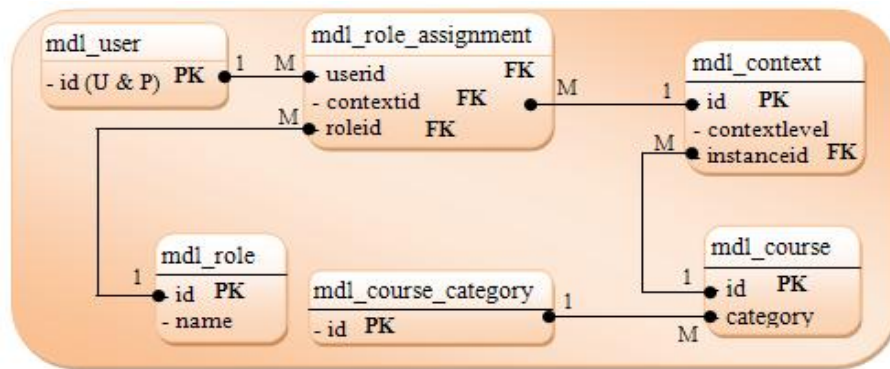


Figure 8.11: The Relationship of the Select Query of searchCourse Service in WSMS

A contextlevel in mdl.context table should have one of the following values; we have chosen 50 in our example because we deal with course as in Figure 8.12.

```
// context definitions
define('CONTEXT_SYSTEM', 10);
define('CONTEXT_PERSONAL', 20);
define('CONTEXT_USER', 30);
define('CONTEXT_COURSECAT', 40);
define('CONTEXT_COURSE', 50);
define('CONTEXT_GROUP', 60);
define('CONTEXT_MODULE', 70);
define('CONTEXT_BLOCK', 80);
```

Figure 8.12: The Context Definitions in Moodle

8.5.7 Login Services

In WSMS, there are three users (admin, teacher and student) who can log in and access their details, depending on the user. Their usernames and passwords are the same as in their own schools. Guest users only need to provide the information that

the service owner specifies in the service parameters, such as the getCourse service that has id, summary and etc. parameters, as in Figure 8.3.



Figure 8.13: Login Form in WSMS

Figure 8.3 displays the complex type of login service. This complex type has two arrays: username and password. Each array has one array as a complex type.

```
// Get User Data Complex Data Type
//*****
$server->wsdl->addComplexType(
    'UserDataInput',
    'complexType',
    'struct',
    'all',
    '',
    array(
        'username' => array('name' => 'username', 'type' => 'xsd:string'),
        'password' => array('name' => 'password', 'type' => 'xsd:string')
    )
);
```

Figure 8.14: The Complex Type for login services

The login service must register, as all services do, in NuSOAP, and the login details (username and password) are in getUserData, as in Figure 8.14.

When users log in, WSMS stores the user details in a session to enable them to browse all the pages that the service owner allows them to see them, as in Figure 8.16. For more security the WSMS destroys the session and all the user details

```
//Check User
$server->register('getUserData', // method name
    array('input'=>'tns:UserDataInput'), // input parameters
    array('return' => 'tns:UserDataOutput'), // output parameters
    'urn:serverwsdl', // namespace
    'urn:serverwsdl#getUserData', // soapaction
    'rpc', // style
    'encoded', // use
    'Get User Data if exists' // documentation
);
```

Figure 8.15: The Registration for login services in NuSOAP

when the user logs out. It is important to obliterate all login data for extra security because these data are already in the database in their school.

```
//Get Input Username and Password
if (isset($_POST["txtUsername"]) && isset($_POST["txtPassword"]))
{
    //Set array items values
    $input['username'] = $_POST["txtUsername"];
    $input['password'] = md5($_POST["txtPassword"]);
    //Set array parameter to call 'getUserData' function at the Web Service
    $param = array('input' => $input);
    if (isset($_POST["School"]))
    {
        //Check which school has been selected
        if ($_POST["School"] == 'ajlan')
            $user_data = $ws_ajlan->call('getUserData', $param);
        else if ($_POST["School"] == 'saudi')
            $user_data = $ws_saudi->call('getUserData', $param);
    }
    //Check whether the return value is set (User found)
    if (isset($user_data))
    {
        //Get array items from the return value
        $_SESSION['userID'] = $user_data['id'];
        $_SESSION['userFirstname'] = $user_data['firstname'];
        $_SESSION['userLastname'] = $user_data['lastname'];
        $_SESSION['userEmail'] = $user_data['email'];
        $_SESSION['userType'] = $user_data['usertype'];
        $_SESSION['School'] = $user_data['school'];
        $page_name = 'search.php'; //session=$_SESSION['userType'];
        header(sprintf("Location: %s", $page_name));
        exit();
    }
    else
    {
        //the return value doesn't exist (User is not found)
        header("Location: login.php?error=1");
        exit();
    }
}
```

Figure 8.16: The Session to Store User's Data

8.5.8 Student Service

As with any service in our system, a student service must register with the server (NuSOAP), as in Figure 8.17. A student service is a complex service, and therefore it has a complex type, as in Figure 8.18.

```
$server->register('getStudents',      // method name
    array('courseid'=>'xsd:int'),
    array('return'=>'tns:ArrayOfStudents'),
    'serverwsdl',                    // namespace
    'serverwsdl#getStudents',        // soapaction
    'rpc',                           // style
    'encoded',                       // use
    'Get all students for a course'   // documentation
);
```

Figure 8.17: The Registration of Student Service in Server (NuSOAP)

```
// Student & ArrayOfStudents Complex Data Types
//*****
$server->wsdl->addComplexType(
    'Student',
    'complexType',
    'struct',
    'all',
    '',
    array(
        'id' => array('name' => 'id', 'type' => 'xsd:int'),
        'username' => array('name' => 'username', 'type' => 'xsd:string'),
        'firstname' => array('name' => 'firstname', 'type' => 'xsd:string'),
        'lastname' => array('name' => 'lastname', 'type' => 'xsd:string'),
        'email' => array('name' => 'email', 'type' => 'xsd:string'),
        'phonel' => array('name' => 'phonel', 'type' => 'xsd:string'),
        'phone2' => array('name' => 'phone2', 'type' => 'xsd:string'),
        'address' => array('name' => 'address', 'type' => 'xsd:string'),
        'city' => array('name' => 'city', 'type' => 'xsd:string'),
        'country' => array('name' => 'country', 'type' => 'xsd:string'),
        'lastlogin' => array('name' => 'lastlogin', 'type' => 'xsd:dateTime'),
        'school' => array('name' => 'school', 'type' => 'xsd:string')
    )
);
```

Figure 8.18: The Complex Type of Student Service

8.6 Summary

This chapter has introduced the proposed approach that is aimed to extend the current architecture of Web services to meet the technical requirements of Moodle as in Chapter 5. It has described the extension of PHP Moodle to support Web services by using the NuSOAP technique. Moreover, it has illustrated that using Web services technology is a straightforward evolution of Web programming models and has demonstrated how PHP can be used as a fast and easy development tool for creating them.

In this chapter, we have provided a comprehensive understanding for developing SOAP services and consumers with PHP. It mentions the most commonly used SOAP implementations used today and focuses on the NuSOAP technique for developing SOAP applications, as well as writing clients to retrieve data from external servers. This has not only given us a good overview of how communication between clients and servers functions, but also how these XML structures get parsed into PHP variables. In addition, this chapter has provided an overview of the most commonly used classes and methods in the WSMS system through the implementation of Web services in order to support PHP Moodle by using the NuSOAP technique.

The next chapter will focus on analysis and evaluation models of VLEs. Then, it will give more details around the value of the research study to research community, teachers and learners. Moreover, it will illustrate the analysis of the WSMS model for VLEs. The last part of next chapter will evaluate models of VLEs using a questionnaire for four users admin, teacher, student and quest.

Chapter 9

Analysis and Evaluation of the WSMS Model for VLEs

Objectives

- To introduce the value of the research study for the research community, teachers and Learners.
 - To illustrate the analysis of the WSMS model for VLEs .
 - To evaluate WSMS model for VLEs including a questionnaire.
-

9.1 Introduction

With the growing reputation of the Web, not only as a research tool, but also as a bona fide business communication tool, groups are finding new ways to communicate with each other. Fortunately, technologies such as Web services are allowing communication structures to be built easily and quickly, allowing even more participation. They assist this process by using the information on the Internet in new

ways, even if only allowing users to subscribe to pouches of information. Services are directly notified when any new information is published, and this helps to create complex business applications and to control complex real-time provide chains [138].

Systems architectures normally become more and more complex through increasing functionality and implementing additional features. Therefore, analysis and evaluation processes are an important process for breaking a complex topic or substance into smaller parts to gain a better understanding of it. This implies a common system view with comparable structures, similar modelling approaches, and computer-aided system evaluation. The overriding process of analysis and evaluation in the WSMS is to gain a greater understanding of the needs of the users in order that the WSMS system is better able to meet those needs in future [178].

The main aim of this chapter is to focus on VLE analysis and evaluation models. It will focus on the value of the research study to the research community, teachers and learners. In addition, it will illustrate the analysis of the WSMS model for VLEs. Finally, this chapter will evaluate VLE models using four closed basic questions and a questionnaire for four users, who are admin, teacher, student and guest.

9.2 The Value of the Research Study

Using the Internet to support teaching and learning has become a trend in modern higher education by using VLEs that support online learning and teaching over the Internet. Therefore, deploying techniques such as description, registration, discovery etc. will open a wide door for collaborative VLE services that are more widely distributed and that operate in a flexible and effective manner. Moreover, VLEs support the worldwide trend of offering online joint courses over the Internet, which includes institutes in different countries. This part will consider the value of this research for the research community, teachers and learners as below:

9.2.1 The Research Community

This research has laid a good foundation for the research community, especially for other researchers intending to work in this area by providing a full picture of Web services, VLEs and their components and related issues. This research has used a complex technology (Web services technology) with a large and important software package (Moodle as an example of VLE). The biggest beneficiary of this research is Higher Education and especially the universities. This study enables many institutions in different countries to work together and share materials that are stored and hosted in different servers by deploying VLE as a service that can be published, located and invoked across the Web. This thesis has provided the following benefits for the research community:

1. It has contributed to the solution of a serious and complex problem that faces institutions all over the world by allowing a number of applications to be integrated between these institutions more quickly, easily and cheaply than ever before.
2. The most important advantage of this research is that it allows educators, students and guests at different institutions in different countries to exchange data and cooperate fruitfully and usefully, and to share material and work together even though the information is stored and hosted in different VLE software platforms, both cheaply and easily.
3. The technology that is used in this study allows developers to separate software engineering from software programming, and to redefine it into a) service providing (for programmers), b) application building (for software engineers), and c) service brokering. This has been done in order to highlight software engineering, and to de-emphasize programming.

- This technology allows pieces of software to be written in different lan-

guages and to run on different operating systems, both cheaply and easily.

- It allows applications running in different parts of an organization and in different organizations to exchange data easily and cheaply.
- It can have applications running anywhere on any technology or device that has Web services.

4. The WSMS system is a multi-language platform as we can see in Figure 9.3.
5. This project has reduced the time and effort needed by offering a good foundation for other researchers intending to work in this area by providing a full picture of Web services, VLEs and their components and related issues.
6. The WSMS system has an excellent login service that enables users to log in to WSMS just as they log in to their own schools. This service checks whether the kind of user exists, and if they exist, it will direct them to their file. Otherwise, it will give them a message ‘Username and/or password are invalid’ and consider them as a guest. After users have logged in, WSMS stores these details in a session to enable users to browse their pages, and after they have logged out, the session will destroy those details.

According to the results of the questionnaire, it has been confirmed that the WSMS system satisfies all kinds of user, especially teachers and students. Most of the users (teachers and students) agree and are very happy with this new idea. In addition, the users’ comments also prove that this project will enhance knowledge, especially in e-learning systems. Moreover, they are satisfied with the login as they can log in with their own username and password without needing to register. The last point is that the authentication and authorization of the system are satisfactory for all users and gives them sufficient trust to deal with this project.

9.2.2 Teachers

Teachers play a major role in any VLE, because teachers are an active element in the educational process. Their position is critical to the future aspirations of the many students that come into the classroom. For example, when schools have good and effective teachers, greater numbers of the students will graduate and will enter higher education.

In addition, teachers play an important role in our WSMS system because teachers can control both their courses and their activities with their students. They can publish their courses and their activities as a service in WSMS so as to be easy and quick for students to discover. The most important advantage of this study is that it allows teachers at different institutions in different countries to work together and share material by connecting individual courses with their activities together, even though stored and hosted on different VLEs. Therefore, teachers at different institutions who teach the same course can share courses modules, assignments, chat, etc. and communicate with each other both cheaply and easily. This thesis has provided the following benefits for teachers:

1. The most important advantage of this study is that teachers can integrate their work with other teachers, even those who are teaching at different institutions in different countries.
2. The WSMS enables teachers to search and discover a set of courses by filtering a number of available courses that register with a WSMS system, and then to select the most suitable ones. This study encourages teachers to discover courses that are similar to their own courses, and to exchange data both easily and cheaply.
3. Teachers do not need to register separately with the WSMS, because the login service in this system enables them to log in as they do to their own schools.

4. The WSMS system allows teachers to teach in two or more schools anywhere in the world.
5. Teachers can log in to WSMS to access all the data for their students. For example, they can see the grades of their students in every module that they teach (in one or more schools), but they cannot see the other modules that they do not teach, because the only one who can see all this information is the admin user.

According to the results of the teacher questionnaire, the majority of teachers are very satisfied; most of them are in agreement and are very happy with WSMS. They said in their comments, ‘this system is useful and will help most teachers around the world to work together in a flexible, widely distributed and effective manner. In addition, it will help students to enjoy any course anywhere in the world’.

9.2.3 Learners

Learners form the body of any VLE and accordingly they play an important role in the WSMS system. This study has contributed to enabling learners to discover the courses that are best suited to them and to find the activities that are most similar to or directly related to those courses. Learners can search a set of semantically equivalent services (course & assignment) by filtering a number of available services that meets their needs, and then selecting the most suitable service during a single execution. This project has thus reduced the time and effort needed by students. In addition, there is better coordination and cooperation between academic institutions than before. This thesis has provided the following benefits for learners:

1. The most important advantage of this study is that learners can discover other courses and related activities that are similar to their own courses. They can also contact their own and other teachers and students, even when those other courses are hosted at different institutions in different countries.

2. Learners do not need to register with the WSMS system because the login service in this system is very straightforward in that it enables learners to log in as they usually do to their own schools.
3. Learners can log in to WSMS to access all the data and related activities for their courses. For example, they can see the grades of their courses in every module that they are enrolled on, but they cannot see the other modules with which they are not enrolled.
4. This system allows learners to work and share with other learners in other institutions if they have only one teacher. For example, if a teacher teaches a module in two universities using the WSMS system, the learners in these universities can work together more easily, quickly and cheaply than ever before.
5. This system furnishes learners some basic information about other teachers and learners, such as name, telephone (home and mobile) and email, to enable these learners to contact them even though these users may be at different institutions and in different countries.

According to the results of student questionnaire, the majority of students are in agreement and are happy with the WSMS system. This system allows them to discover other courses and related activities, and to work with other students and even other teachers in different institutions in different countries anywhere in the world. In addition, it enables them to gather more information relating to their courses, or at least similar ones, in an easy, cheap and quick way. They said in their comments, ‘if we use this system, it will help us to work together with others students and also teachers and discuss our issues with each other. Also, it will allow us to benefit from any course anywhere in this system over the world’.

9.3 Analysis of Models for VLE

The architecture of the WSMS system, as in Figure 6.12 in Chapter 6, lays the conceptual foundation for establishing interoperable Web services in VLE. It identifies a number of important abstractions and their interdependencies. The contribution of this architecture provides a coherent framework that allows specific technologies to be considered in a logical context and facilitates the work of specification writers and architects [179]. In this section, we will analyse the most important services in our WSMS system in order to measure the quality of these services. The analysis will examine them and explain the function of each service and the relationships between them.

9.3.1 Course and Assignment Services

At the moment, the WSMS system has 10 services and the main ones are course and assignment services. One of the main targets of this system is to enable users to search and find the most suitable course and related activities in all schools that have registered with WSMS. This system enables users to see more details about the courses; as the owner of each service (school) had specified them when they published them in WSMS. The current details of the courses that have been published by both Saudi School in Leicester (SSiL) and Ajlan's High School (AHS) are displayed in Figure 9.1.

As we can see in Figures 9.1 and 9.3, the result of search is 14 courses with their details (name, category, start date, assignments, students and school). In addition, users can search by assignment for all schools or can choose a specific school, as in the middle of Figure 9.1.



Course	Category	Start Date	Assignments	Students	School
تاريخ - للصف الاول ثانوي	الصف الاول الثانوي	15/04/2008	2	12	Ajlan's High School
تاريخ - للصف الاول متوسط	الصف الاول المتوسط	01/04/2009	4	19	Ajlan's High School
تاريخ - للصف الثالث متوسط	الصف الثالث المتوسط	15/07/2008	1	13	Ajlan's High School
تاريخ - للصف الثاني متوسط	الصف الثاني المتوسط	16/02/2008	6	9	Ajlan's High School
تاريخ للصف السادس	الصف السادس ابتدائي	15/03/2008	2	19	Ajlan's High School
تاريخ للصف الاول ثانوي	الصف الاول الثانوي	15/10/2007	0	12	المدرسة السعودية بلستر
تاريخ للصف الاول متوسط	الصف الاول المتوسط	02/01/2008	2	19	المدرسة السعودية بلستر
تاريخ للصف الثالث متوسط	الصف الثالث المتوسط	15/10/2007	3	13	المدرسة السعودية بلستر
تاريخ للصف الثاني متوسط	الصف الثاني المتوسط	15/10/2007	1	9	المدرسة السعودية بلستر
تاريخ للصف الخامس ابتدائي	الصف الخامس ابتدائي	15/10/2007	5	27	المدرسة السعودية بلستر

Figure 9.1: The Main Page of WSMS where all users can search and see basic information

9.3.2 Main Page

The main page in WSMS is very simple, like the Google main page. The scenario of WSMS starts with the request of the (potential) student who types his/her course/assignment in the main page, as in Figure 9.2. It has the ability to find the optimal services that meet that user's needs. WSMS will search in NuSOAP as an x-UDDI registry to check whether or not the service is registered. Each service has a select database, as in Figures ref Database9 and 8.10 in Chapter 8, which enables WSMS to obtain service details, depending on the service owner's specifications (Provider 1 & Provider 2) as in Figure 6.12 in Chapter 6, in order to locate the requested service and bind it to the user.

As in most VLE systems, there are four users, which are admin, teacher, student and guest. The authorization in WSMS is divided into four levels, as in Figure 5.9 in Chapter 5. Before logging in, all users can search and see their details, and view



Figure 9.2: The Front Page of WSMS

basic information about the courses in their school or in other specified schools, as in Figure 9.1.

9.3.3 Login Service

As we explained about this service in Chapter 8, admin, teacher and student users have the ability to log in to WSMS just as they do to their own schools. WSMS has a service that checks whether the kind of user exists, and if they exist, it will direct them to their file. Otherwise, it will give them a message ‘Username and/or password are invalid’ and consider them as a guest. After users have logged in, WSMS stores these details in a session to enable users to browse their pages, and after they have logged out, the session will destroy those details.

9.3.4 Admin Service

Every user in WSMS system has his/her own services. These services enable users to see all the details that the services’ owner (school) has registered and published in the NuSOAP registry. The WSMS system gives full permission for admin to see all the details in his/her school, as in Figure 9.3.

Welcome: **عن ناصر حسن عبدالوهاب**
 Class: **الصف الاول الثانوي**
 School: **المدرسة السعودية بلسنر**

Administrator: 1
Student: 254
Teacher: 15

Ajlan Moodle Services

Please Type-in some text to search for existing courses:

تاريخ

☒ Course ☐ Assignment

- All Schools -

Search

Results (14)

Course	Category	Start Date	Assignments	Students	School
تاريخ - للصف الاول ثانوي	الصف الاول الثانوي	15/04/2008	2	12	Ajlan's High School
تاريخ - للصف الاول متوسط	الصف الاول المتوسط	01/04/2009	4	19	Ajlan's High School
تاريخ - للصف الثالث متوسط	الصف الثالث المتوسط	15/07/2008	1	13	Ajlan's High School
تاريخ - للصف الثاني متوسط	الصف الثاني المتوسط	16/02/2008	6	9	Ajlan's High School
تاريخ للصف الخامس	الصف الخامس ابتدائي	15/08/2008	5	27	Ajlan's High School
تاريخ للصف الرابع ابتدائي	الصف الرابع ابتدائي	15/10/2007	3	29	Ajlan's High School
تاريخ للصف السادس	الصف السادس ابتدائي	15/03/2008	2	19	Ajlan's High School
تاريخ للصف الاول ثانوي	الصف الاول الثانوي	15/10/2007	0	12	المدرسة السعودية بلسنر
تاريخ للصف الاول متوسط	الصف الاول المتوسط	02/01/2008	2	19	المدرسة السعودية بلسنر
تاريخ للصف الثالث متوسط	الصف الثالث المتوسط	15/10/2007	3	13	المدرسة السعودية بلسنر
تاريخ للصف الثاني متوسط	الصف الثاني المتوسط	15/10/2007	1	9	المدرسة السعودية بلسنر
تاريخ للصف الخامس ابتدائي	الصف الخامس ابتدائي	15/10/2007	5	27	المدرسة السعودية بلسنر

Figure 9.3: The Main Page of Admin, Teacher and Student Users

The difference between Figure 9.1 and Figure 9.3 is the blue link on the Course, Assignments and Students columns. Admin, teacher and student users can access these links after login. The difference between admin and student is that admin can see all the data, as in Figure 9.4, and student can only see some of the data, as the service owner has specified, as in Figure 9.5.

Back Home Logout

Students (12)

ID	First name	Last name	Address	Phone	Username	Email	Last Login
248	أفان	فوزي	SA سعودي.انتي منتظمه	SA سعودي.انتي منتظمه	st310507	st310507@yahoo.com //	
247	اسراء	يوسف محمد خليفة	LY لبييه انتي منتظمه	LY لبييه انتي منتظمه	st310407	st310407@yahoo.com //	
253	المدينق	خالد ابو القاسم سويبي	LY لبييه نكر منتسب	LY لبييه نكر منتسب	st311007	st311007@yahoo.com //	

Figure 9.4: The Student Details that Admin can see

Students (12)			
ID	First name	Last name	Address
248	أفنان	فوزي	SA, سعوديہ انتی منتظمہ
247	اسراء	يوسف محمد خليفة	LY, ليبيا انتی منتظمہ
253	الصادق	خالد ابوالقاسم سويسی	LY, ليبيا ذکر منتسب

Figure 9.5: The Student Details that Student can see

9.3.5 Teacher Service

Teachers are important users and can log in to WSMS to access all the data for their students. For example, they can see the grades of their students in every module that they teach (in one or more schools), but they cannot see the other modules that they do not teach, because the only one who can see all this information is the admin user.

9.3.6 Student Service

A student service, as in the last section, enables students to see all their courses and assignments (around the world) as long as they have been registered with WSMS. This service displays important information for students, such as grades, and this information depends on the decision of the service owner (school). WSMS allows students to see only their grades, and if students try to see other students' grades, the system gives them the message, 'It is either no students have taken this assignment and/or you are not permitted to see this grade', as in Figure 9.6.

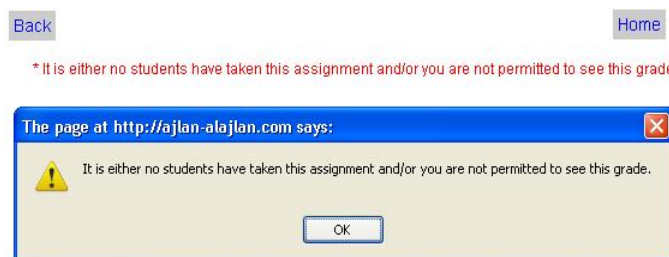


Figure 9.6: The Message that appear for users if they do not have access permission

9.3.7 Guest Service

Guest users do not have a login service and can only see some limited details about courses such as names, assignments, students and schools, as in Figure 9.1. If a guest requires more information about any course or assignment, they can visit that school by clicking on the name of the school; WSMS provides guests with a link to go directly to any selected school.

9.4 Evaluate the Models for VLE

Evaluation is the process of examining a subject, and rating is based on its significant features. We decide how much or how little we value something, using our judgement but depending on criteria that have been predefined. In this section, several important questions will be asked, and the answers to these questions will provide explanations for the main aspects of WSMS [178].

Q1. Is this system secure?

It is important that the data can be trusted before being processed, and WSMS uses languages that are very secure. These languages are XML, WSDL, SOAP and PHP. WSMS needs to be sufficiently secure so that institutions can trust publishing their services all over the world. WSMS has divided users into four levels, as in Figure 5.9 in Chapter 5. The authentication of our system has the ability to check the following points:

1. Check if users exist in any school (SSiL and AHS).
2. Check and audit the kinds of users and direct them to their pages.
3. Users do not need to register; they can log in by using the username and password from their own school.

4. The system gives users more confidentiality and privacy; it does not allow any user to see the data of other users. For example, students cannot access teacher or admin services.

Q2. Is this system easy or hard to use?

According to the questionnaire, we can see that the system is easy and simple to use, and has a friendly interface. The main page is a simple page and it is similar to the Google main page. Users can search for any course and related assignment in any school that is registered with this system without logging in. In this situation, users can access information about courses such as name, category, assignment, student and school. On the other hand, admin, teacher and student can log in and search to obtain more details about schools and their activities, depending on the kind of user, as in Figure 5.9 in Chapter 5. In addition, WSMS offers extra information for users who can log in, such as the number of administrators, teachers and students in every school, as in the left top of Figure 9.3.

Q3. Does this system meet overall goals?

The results of the questionnaire confirm that users are happy with WSMS, especially teachers and students. The main point that users are satisfied with is that this project opens a wide door for collaborative VLE services that are flexible, widely distributed and effective. Furthermore, VLEs support the worldwide trend of offering online joint courses over the Internet, which includes institutes in different countries who employ roaming staff and target mobile students.

Q4. Do you think that students and teachers benefit from this system?

This system enables teachers and students who are using Moodle in different institutions in different countries around the world to work together and share material, although that material may be hosted on different VLEs.

9.4.1 Questionnaire

Questionnaires are generally viewed as a valid method of assessment, and ours are capable of providing significant discrimination between the various types of users with respect to WSMS system skills. The questionnaires are based on samples of administrators, teachers, students and guests. The answers to the questions are a choice of: strongly agree, agree, normal, disagree, and strongly disagree. The questions depend on the kind of user and we have tried to set these questions to consider that kind of user. Each of these groups will now be considered as below:

9.4.1.1. Guest Questionnaire

The purpose of the questionnaire for guests is to discover their impression about WSMS system, and also to get their feedback. The study has chosen 46 guests and the results of their answers are summarized graphically in the following figures:

1. What is your impression of the site?

The answer to this question is that 8 guests strongly like the design of system and most of them (22) said it is normal; 14 like it and 6 dislike it. Finally, only 2 guests strongly dislike this system.

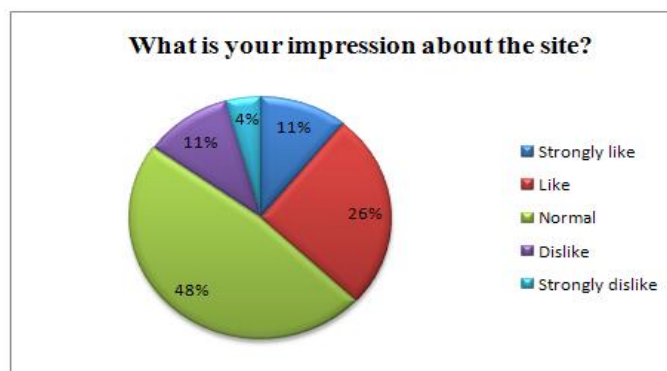


Figure 9.7: The results of answers of Question 1 by guests

2. Please type 'History' in search form and press search.**a) In general, are you satisfied with the result?**

The answer to this question is that 13 guests strongly like the design of the system and most of them (18) said they agree; 9 think it is normal and 5 disagree. Finally, only 1 guest strongly disagrees.

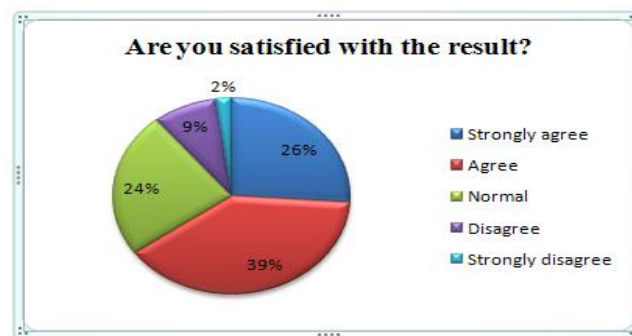


Figure 9.8: The results of the answers to Question 2-a by guests

b) Are you satisfied with the interface of the result?

The answer to this question is that 5 guests strongly agree with the design of the system and 12 of them said it is normal; 14 agree and 11 disagree. Finally, 4 guests strongly disagree.

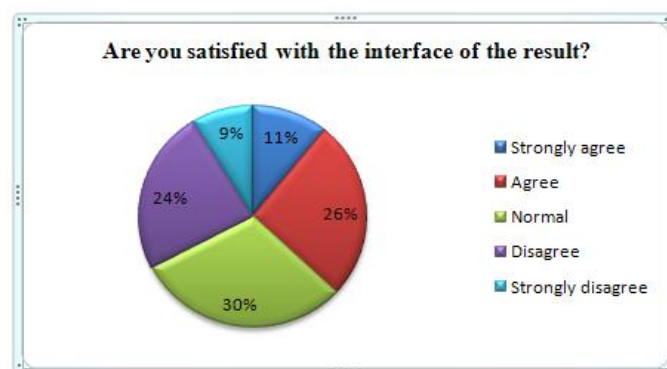


Figure 9.9: The results of the answers to Question 2-b by guests

c) Are you satisfied with the use of this system?

The answer to this question is that 6 guests strongly agree with the design of the system and 22 of them said it is normal; 10 agree and 6 disagree. Finally, only 2 guests strongly disagree.

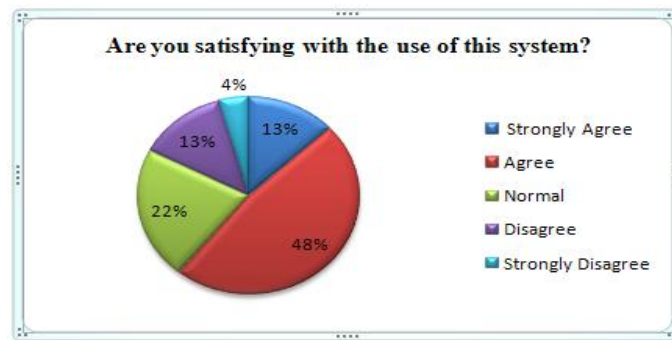


Figure 9.10: The results of the answers to Question 2-c by guests

Overall, the results of the questionnaire for guests are that users are satisfied and most of them agree and are happy with this system although they do not have login and can only search to view general information about courses in the schools that publish their services.

9.4.1.2. Student Questionnaire

The purpose of the questionnaire for students is to discover their impressions of the WSMS system, and also to get their feedback. The study has selected 130 students out of 291 from SSiL. The students in this school are from various countries such as Saudi Arabia, Libya, UK, Somalia, Syria, Palestine, Egypt, Algeria, Morocco and the Gulf countries. The results of the answers to the questions are summarized graphically in the following figures:

1) What is your impression of the site?

The result is that 9 students strongly like a system and most of them (56) said it is normal; 28 like it and 26 dislike it. Finally, 11 students strongly dislike a system.

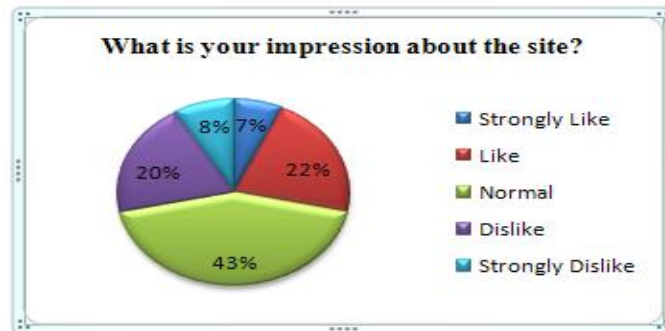


Figure 9.11: The results of the answers to Question 1 by students

2) Please log in to the site using your own username and password, and type 'History' in the search form and click search.

a) In general, are you satisfied with this result?

The result is that 18 students strongly agree and 47 of them said it is normal; 42 agree and 15 disagree. Finally, 8 students strongly disagree with this system.

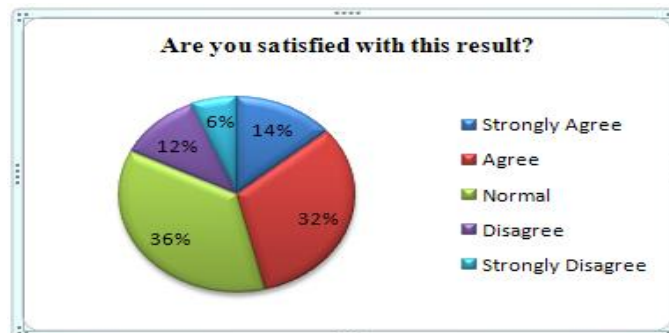


Figure 9.12: The results of the answers to Question 2-a by students

b) Are you satisfied with the interface of the result?

The result is that 21 students strongly agree and 49 of them said it is normal; 44 agree and 12 disagree. Finally, 4 students strongly disagree with this system.

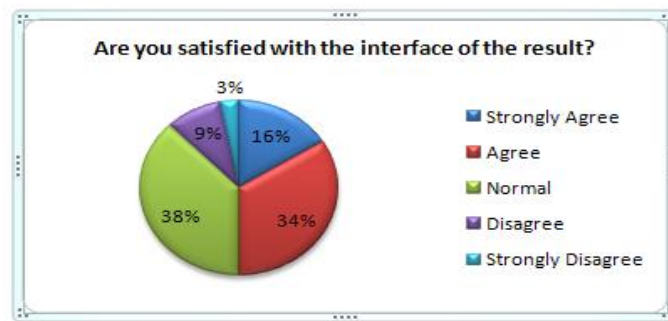


Figure 9.13: The results of the answers to Question 2-b by students

c) Are you satisfied with the use of this system?

The answer to this question is that 14 students strongly agree with the design of the system and 41 of them said it is normal; 32 agree with it and 25 disagree. Finally, 8 students strongly disagree with this system.

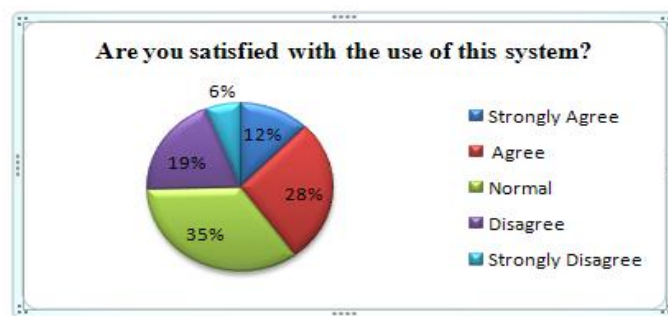


Figure 9.14: The results of the answers to Question 2-c by students

3) Please log in to the site using your username and password and search for any of your courses, and then click on the assignment and the press on the name of the assignment to see your grade:

a) In general, are you satisfied with the assignment details?

The result is that 24 students strongly agree and 34 of them said it is normal; 54 agree with and 13 disagree. Finally, 5 students strongly disagree with this system.

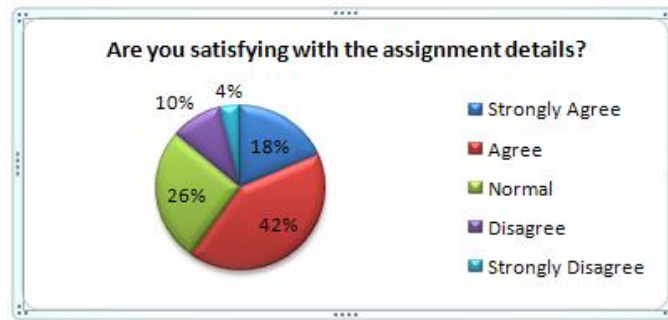


Figure 9.15: The results of the answers to Question 3-a by students

b) Are you satisfied with seeing your grade on the chosen course?

The result is that 29 students strongly agree and 31 of them said it is normal; 57 agree and 10 disagree. Finally, 3 students strongly disagree with this system.

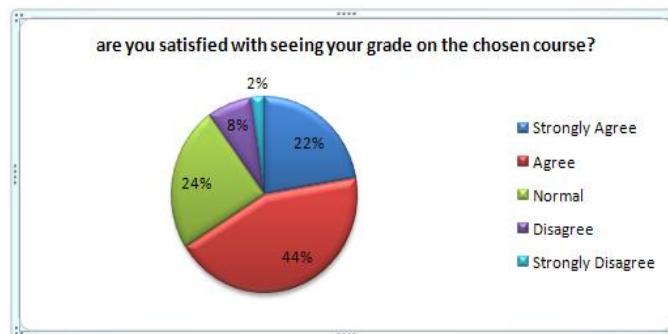


Figure 9.16: The results of the answers to Question 3-b by students

Overall, the results of the student questionnaire show that the majority of them are satisfied, especially with the student login. Most of them agree and are happy with this system and, as they said in their comments (Appendix G), 'this website will help us to discover all courses anywhere in the world, especially those that are similar to our courses and assignments' and, 'it is an easy and fast way to see our grades and check our assignments'.

9.4.1.3. Teacher Questionnaire

The purpose of the questionnaire for teachers is to discover their impressions of the WSMS system, and also to get their feedback. The study has selected 18 teachers from SSiL. The teachers in this school are also from various countries such as Saudi Arabia, Libya, UK, Syria, Palestine, Egypt, Algeria, Morocco and the Gulf countries. The results of the answers to the questions are summarized graphically in the following figures:

1) What is your impression of the site?

The result is that 3 teachers strongly like this system and 4 of them said it is normal; 7 like it and 3 dislike it. Finally, only 1 teacher strongly dislikes this system.

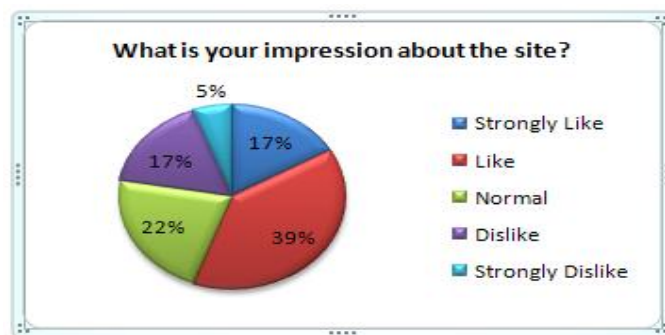


Figure 9.17: The results of the answers to Question 1 by teachers

2) Please log in to WSMS using your username and password and type 'History' in the search form and click search.

a) Are you satisfied with the result?

The result is that 2 teachers strongly agree with the system and 8 of them said it is normal; 5 agree with it and 2 disagree. Finally, only 1 teacher strongly disagrees with this system.

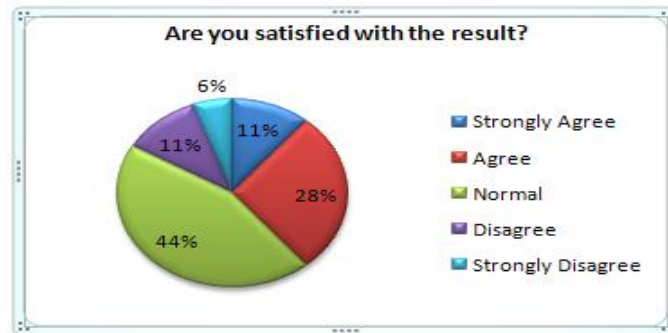


Figure 9.18: The results of the answers to Question 2-a by teachers

b) Are you satisfied with the interface of the result?

The result is that 2 teachers strongly agree with the system and 5 of them said it is normal; 7 agree with it and 3 disagree. Finally, only 1 teacher strongly disagrees with this system.

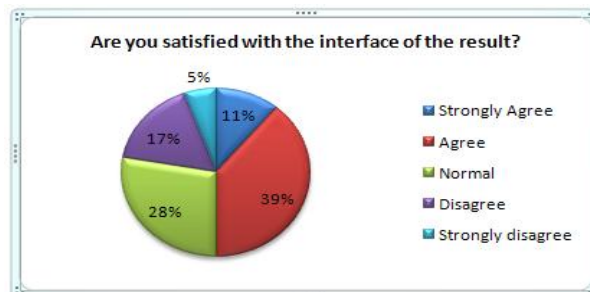


Figure 9.19: The results of the answers to Question 2-b by teachers

c) In general, are you satisfied with seeing your courses and students in this system?

The result is that 4 teachers strongly agree with the system and 6 of them said it is normal; 6 agree with it and 1 disagrees. Finally, only 1 teacher strongly disagrees with this system.

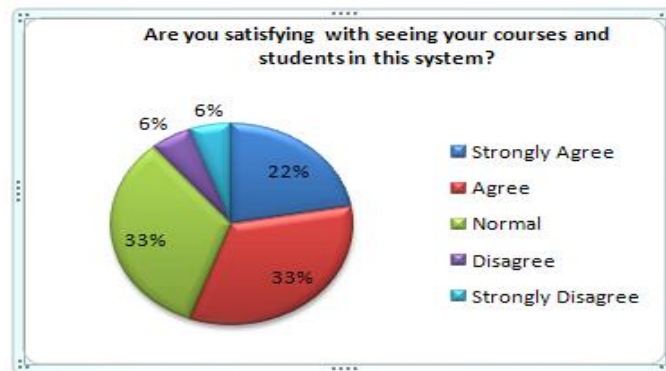


Figure 9.20: The results of the answers to Question 2-c by teachers

3) Please log in to the site using your username and password and search for any of your courses, and then click on the assignment and the press on the name of the assignment to see your student's grades:

a) Are you satisfied with the course and assignment details?

The result is that 4 teachers strongly agree and 6 of them said it is normal; 5 agree with and 2 disagree. Finally, 1 teacher strongly disagrees with this system.

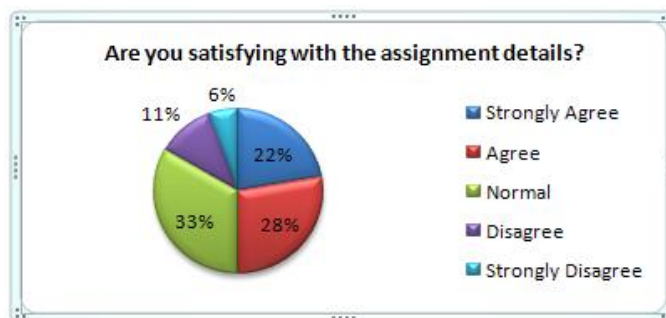


Figure 9.21: The results of the answers to Question 3-a by teachers

b) Are you satisfied with seeing your grade on the chosen course?

The result is that 3 teachers strongly agree and 5 of them said it is normal; 6 agree and 3 disagree. Finally, only 1 teacher strongly disagrees with this system.

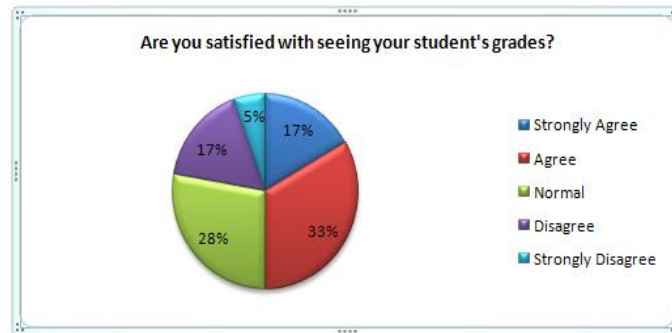


Figure 9.22: The results of the answers to Question 3-b by teachers

Teachers are important users in our system because they are at the heart of VLE and they have the ability to control courses and students. Generally, the results of teacher questionnaire show that the majority of them are very satisfied. Most of them agree and are very happy with WSMS. They said in their comments, ‘this system is useful and will help all teachers around the world to work together in a flexible, widely distributed and effective manner. In addition, it will help students to enjoy any course anywhere in the world’.

9.4.1.4. Administrator Questionnaire

The purpose of the questionnaire for administrators is to discover their impressions of the WSMS system, and also to get their feedback. The study has only 6 administrators from both AHS and SSiL. The administrators in these schools are all from Saudi Arabia because these schools are supported by the Saudi government. The results of the answers to the questions are summarized graphically in the following figures:

1. What is your impression of the site?

The result is that 1 administrator strongly likes this system and 2 of them said it is normal; 2 like it and 1 dislikes it. Finally, no administrator strongly dislikes this system.

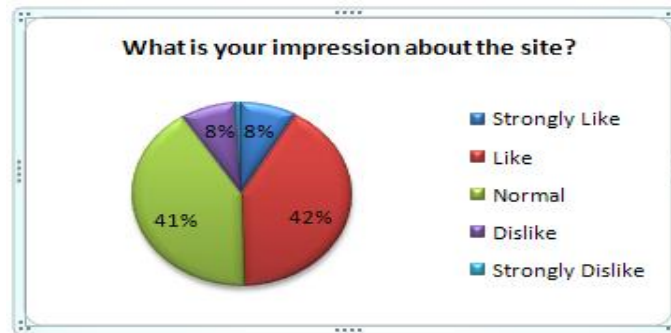


Figure 9.23: The results of the answers to Question 1 by administrators

2. Please log in to WSMS using your username and password, and type 'History' in the search form and click search.

a) Are you satisfied with the result?

The result is that 1 administrator strongly agrees with the system and 1 of them said it is normal; 3 agree with it and 1 disagrees. Finally, no administrator strongly disagrees with this system.

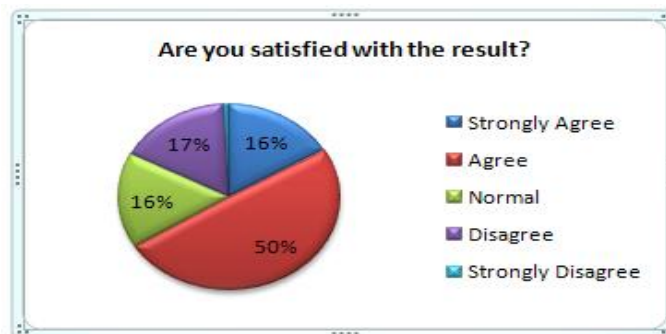


Figure 9.24: The results of the answers to Question 2-a by administrators

b) Are you satisfied with the interface of the result?

The result is that 1 administrator strongly agrees with the system and 2 of them said it is normal; 1 agrees with it and 2 disagree. Finally, no administrator strongly disagrees with this system.

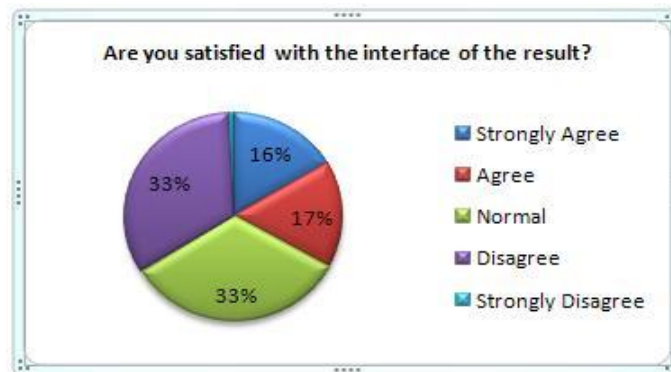


Figure 9.25: The results of the answers to Question 2-b by administrators

c) Are you satisfied with seeing your courses and students in this system?

The result is that 2 administrators strongly agree with the system and 1 of them said it is normal; 1 agrees with it and 2 disagree. Finally, 1 administrator strongly disagrees with this system.

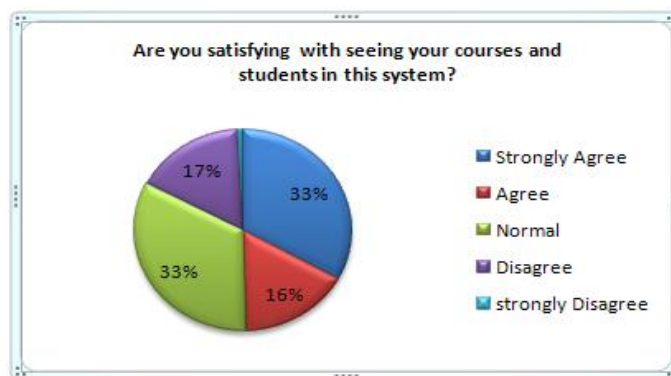


Figure 9.26: The results of the answers to Question 2-c by administrators

The results of the administrator questionnaire were not bad. Some of them agree and are happy with WSMS, but others said that although this is a good idea, it needs a great deal of effort, cooperation and consideration between institutions across the world (Appendix G has some of their comments).

9.4.1.5. The Final Result

Generally, the results of the questionnaires have confirmed that the WSMS system satisfies all four kinds of user, especially the teachers and students as in Table 9.1. Most of the users (teachers and students) agree and are very happy with this system. In addition, their comments also prove that this project will enhance knowledge, especially in VLE systems such as the Moodle system. Moreover, they are satisfied with the login as they can log in with their own username and password without needing to register. The last point is that the authentication and authorization of the system are satisfactory for all users and gives them sufficient trust to deal with this project.

Table 9.1: The Final Result for all Questions of the Questionnaire for all Users

Q	User	Strongly Agree	Agree	Normal	Disagree	Strongly Disagree
1	Admin	1	2	2	1	0
	Teacher	3	4	7	3	1
	Student	9	28	56	26	11
	Guest	8	14	22	6	2
2-a	Admin	1	3	1	1	0
	Teacher	2	5	8	2	1
	Student	18	42	47	15	8
	Guest	13	9	18	5	1
2-b	Admin	1	1	2	2	0
	Teacher	2	7	5	3	1
	Student	21	44	49	12	4
	Guest	5	14	12	11	4
2-c	Admin	2	1	1	2	1
	Teacher	4	6	6	1	1
	Student	14	32	41	25	8
	Guest	6	10	22	6	2
3-a	Teacher	4	5	6	2	1
	Student	29	57	31	10	3
3-b	Teacher	3	6	5	3	1
	Student	24	54	34	13	5
Total		166	344	375	149	55

Table 9.1 summarises all users' answers in this questionnaire. It proves that the most of users are happy as in columns Agree and Normal. As we can see in all

questions, most of the users (teachers and students) agree and are very happy with this system.

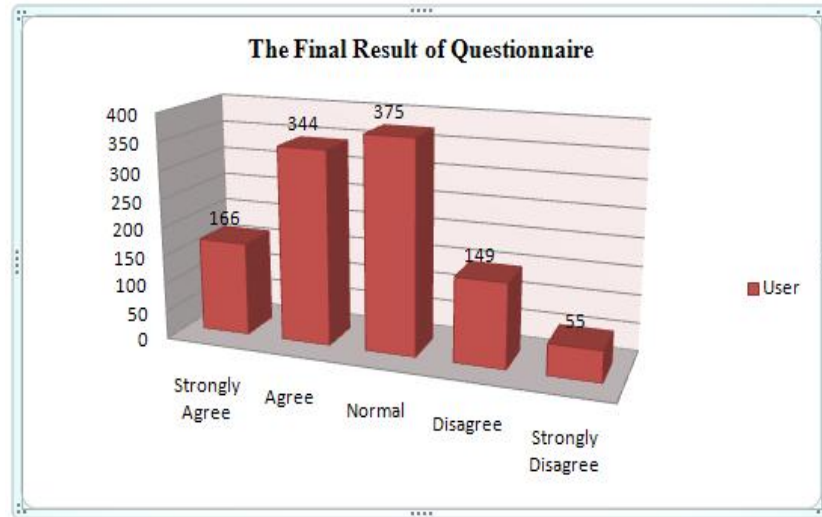


Figure 9.27: The Final Result of the Questionnaire for all Users

Figure 9.27 displays the final questionnaire results for all users. There are 166 users who strongly agree with this system, and in contrast, there are just 55 who strongly disagree. In addition, there are 344 who agree with this system, but only 149 who disagree. Finally, there are 375 who said that this system is normal.

9.5 Summary

This chapter has analysed and evaluated the final models of VLE with the Web services approach. In the beginning of this chapter, we considered the value of this research for the academic community, teachers and learners. In addition, this chapter has analysed WSMS as proposed in this study. The security requirements of VLE systems based on Web services have also been considered in this chapter. Also, the authorization, verification and authentication of security issues have been examined in this chapter.

In the last part of this chapter, we evaluated our WSMS system through a set of basic but important questions and answers, as well as introducing the questionnaires. These questionnaires are based on samples of administrators, teachers, students and guests. The answers to the questions are a choice of: strongly agree, agree, normal, disagree, and strongly disagree. The questions depend on the kind of user and we have tried to set these questions to consider that kind of user.

The next chapter will summarise the work presented in this thesis. The significance of the main findings will be presented in that chapter. In addition, it will highlight the most important contributions made, and will then discuss methods and directions for possible future studies.

Chapter 10

Conclusion & FutureWork

This thesis has been focusing on two important fields for academic institutions across the world, which are VLE and SOC. VLEs are increasingly becoming a significant part of the strategy for delivering on-line and flexible learning, and therefore, there is an increasing demand for their methodologies and technologies. This thesis has presented the key points about SOC: that it involves extended, loosely coupled activities among several independent e-learning partners. Web services, as a part of SOC, is fast becoming an important technology in the evolution of both distributed computing and the Web throughout the world. Web services allow many applications to be integrated faster, more easily and more cheaply than ever before.

The main challenge of this investigation has been to find a suitable solution for using Web services with VLEs and related technologies. Web services allow educators at different institutions in different countries to work together and share material by connecting individual courses together, even though yhy may be hosted on different VLE platforms. This approach has provided a richer social context for VLEs and a set of robust core capabilities that enable instructors to efficiently manage courses, author content, create assignments, and foster collaboration, among other key functions. Moreover, this study has suggested a new approach that meets

this challenge, and it has explored novel architectural designs for VLEs based on the SOC paradigm. This approach has demonstrated that VLEs can be considered as services that can be published, discovered and composed, as perceived within the SOC paradigm [180].

This investigation contains a survey of VLE and SOC methods, a description of the work needed to be done to satisfy the research problem, and an overview of the results and the contributions we have made. It has successfully achieved the proposed work, which was to extend the core idea behind VLE tools by building a VLE around Web services and its related techniques. In addition, this research has answered all the questions that we had posed at the beginning of this thesis, and these questions and their answers are as follows:

1) How can we enable Web services to facilitate the process of building VLE around SOC?

This thesis has succeeded in using Web services, as part of SOC, in tandem with VLE software packages. It has used two VLE systems, which are AHS and SSiL, as in Chapter 8. These VLEs use the PHP language and because there is no standard API to develop Web services applications, we adopted some techniques to help us develop Web services in order to make our VLE systems work smoothly. At the moment, the most popular techniques are NuSOAP, ezSOAP, PEAR::SOAP and NuSphere, but in this research, we have chosen the NuSOAP technique for the reasons mentioned in Chapter 8 [180].

One of the greatest advantages of Web services is that they allow a number of applications to be integrated more quickly, easily and cheaply than ever before. They are expressed as WSDL, which is an XML-based language. A service specifies a contract between the client's requests and the operations that it can offer. A service may be published and discovered using UDDI, while SOAP allows vendor-neutral

communication between applications over HTTP.

The proposed approach has extended the current architecture of Web services to meet the technical requirements of Moodle, as in Chapter 5. This approach allows institutions to exchange data cheaply and easily, and deploys techniques such as service descriptions, registrations, discovery and binding for collaborative VLE services that are distributed, in a flexible and effective manner. Figure 6.12 in Chapter 6 explained how we introduced the Web services approach to meet our VLEs (AHS and SSiL). The most important advantage of this approach is that it allows educators, students and guests at different institutions in different countries to share material and work together through connecting individual courses together, and hosting them at AHS and SSiL servers.

2) How to enhance the traditional security requirements and modify them to cater for the highly mobile and changeable environment of VLE?

Security is an important issue in all fields, especially in e-banking and e-learning. It is a critical issue that every developer must be concerned with when building applications. It is important that the data can be trusted before being processed. The most important area for security and privacy in VLE systems is policy including all types. In this thesis, we propose to use Ponder as an environment for policy and implementation in VLE systems.

This thesis recommends using access control policies with VLE that are used to identify the accessing right for subjects to execute actions on specified objects. These policies protect the resources and services in VLE products. Chapter 7 has four policy types that support access control, which are authorization policy, information filtering policy, delegation policy and refrain policy.

Authorization policy has positive and negative aspects; the positive is used to

specify the action that subjects are permitted to achieve on a targeted object, and the negative is to state the actions that subjects are not allowed to perform on a targeted object. Filtering policies transform the data input/output parameters in an action, for example, a location service may only authorize access to detailed position information, a person in an exact area, or to browsers within the sub-division. Outside browsers can only determine whether someone is at work or not. Delegation policy is used to provide permission to the subjects (through an authorization policy) to delegate all or some of their access rights to a new set of subjects. Refrain policies indicate to a subject to refrain from doing something, and are similar to negative authorization policies but are interpreted by the subject.

The security policies are used to protect the VLE products, to protect them from unauthorized access outside and inside system. These policies are implemented using obligation policy in Ponder as in Chapter 7. Security specifies how well a service provides confidentiality and non-repudiation by authenticating the parties involved, encrypting messages, and maintaining access control. A service provider can provide different security levels and mechanisms depending on the service requester, as in Figure 5.5 in Chapter 5. The WSMS security framework has to enable the some points that are Authentication, Authorization, Confidentiality, Integrity and Auditing (we explained them in more detail in Chapter 5).

WSMS should have robust levels of security; sufficiently strong enough for all schools that deal with a WSMS around the world to trust it. Exchanging data over the Internet can be unsafe and prone to fraud or being lost, and we are therefore increasing our efforts to combat fraud on the system with a variety of requirements. We are also making sure our disclosures about cooperating with law enforcement bodies allow us to respond to increasingly sophisticated fraudsters and criminals. We are using messages to authenticate users if, for example, they are using an unfamiliar computer in order to prevent fraudsters from hacking into the system.

3) What is the best method of supporting the worldwide trend to offer joint online courses over the Internet, which includes institutes in different countries who employ roaming staff and target mobile students?

Currently, the widespread use of Information and Communication Technology, together with VLE software, to support higher education institutions has been increasing all over the world. This is the driver behind plans that wish to see VLEs become fully embedded into e-learning and teaching practice, instead of just using VLEs in their traditional supportive role.

There are more than 250 providers of commercial e-learning and more than 45 of them are OSS VLEs. Of these, the more well known are Moodle, Claroline, SAKAI, WebCT. One of the best VLEs to have emerged to meet the growing interest in OSS is Moodle. It is a free OSS which means users are free to download, use, modify and even distribute it under the terms of GNU. We chose this software to be the focus of our research for the reasons mentioned in Chapter 4.

VLEs are yet fully developed or as widely deployed as they could be. The most important challenges guiding their further development are to retain learners within education, and to encourage a wider group of learners to engage in higher education. One of the approaches taken to meet these challenges is to use Web services to encourage closer working and cooperation between higher education institutions. As we mentioned in the answer to Question 1, the best approach is to support the world-wide trend in offering joint online courses over the Internet, using Web services and benefiting from their advantages to support e-learning.

4) Can this approach be applied to all VLEs, and especially to Moodle?

The answer of this question has two parts, which are:

The first answer is yes, this approach is applicable for VLE (Moodle), and this investigation has proved that using Web services with VLE (Moodle) is possible, as in Chapters 6 and 8. It enables teachers who are using Moodle in different institutions or in different countries around the world to work together and share material, although that material may be hosted on different VLEs, as in Chapter 8. In addition, this approach is applicable to other VLEs that use the PHP language.

The second answer to this question is that we are not sure that the VLE software packages that use other languages, such as Blackboard (V6.2) built in Java, can use this approach. This issue will be considered and confirmed in our future studies.

6) And what are the effects by applying Web services on VLEs (Moodle)?

This thesis has discovered that it has been not easy to work with Web services and its associated technologies using PHP, especially with the release of PHP 5. The inclusion of a diversity of XML tools supplies developers with a store of tools to undertake virtually any type of challenge involving XML. In addition, the latest version of PHP has taken the extra step with the SOAP extension, supporting SOAP servers and clients, allowing developers to create Web services more easily and quickly. Therefore, PHP has become a more feasible solution for implementing applications with XML and Web services but it is still difficult for inexperienced developers to understand how to start using any of these tools. Developers not only need to understand the API of these extensions, but they also need to know which extension they should use. In addition to this, they also need to have an understanding of the specifications of Web services technologies.

The PHP language has strong tools that support security, as well as providing many mechanisms to deal with XML documents, but the security and authentication of data is a big and very important challenge, especially for large VLE software. This

thesis has covered the more important standards and methods for supporting Web services encryption and digital signatures in documents, as in Chapter 9 [60].

7) How can we demonstrate that VLEs may be considered as services that can be fully utilised and published within the SOC paradigm?

As we mentioned in the answer to the above question, it has not easy to work with Web services and their related techniques (XML, WSDL, SOAP, UDDI and HTTP), and to create services and publish them in a service registry in order to enable service requestors to find them cheaply and easily. As in Chapters 2 and 8, the scenario for Web services is that service providers publish and deploy services in a service registry, and a service requestor works with a service broker to discover services; the requestor then negotiates with the service providers to bind those that meet the needs of the requestor. A service requesters could be a human client, a device, an application, or any other Web service. A service broker provides registries for exposing Web services. The roles of service provider and service requestor are logical constructs and a service can exhibit characteristics of both.

Chapters 5, 6 and 8 explained the above approach as we used it in this study. Moodle is written in PHP but currently there is no standard API for developing Web services applications with PHP language. Therefore, we used the most popular technique (NuSOAP) to aid us in developing Web services with Moodle. Chapter 8 has more explanation about this approach and how to apply it.

10.1 Future Work

Using the Internet to support learning has become a trend in modern higher education institutes. In order for current and future generations of personalized VLEs to improve educational effectiveness and efficiency, there are fundamental requirements

that must be realized. While recognizing that the world at large will continue to use terminology in different and often ambiguous ways, the term of VLEs is used to refer to the online interactions of a variety of kinds, which take place between students and instructors. VLEs refer to the components through which students and teachers participate in online interactions of various kinds, including online learning [171].

The approach in this thesis focuses on a Web services technology that support the integration of software applications in an incremental way, using existing platforms and languages that utilize and adopt existing legacy systems. This thesis has succeeded to introduce a Web services technology to meet the requirements of VLE by extending the existing architecture of VLE. The big challenge of our model in the future is to improve the level of security and privacy of the environment of VLE system. Figure 10.1 suggests a Collaborative E-learning Model that has an integration modular security with partners e-learning as a provider service using a Web services technology. This model will apply the introduced approach in this thesis as in Figure 10.1.

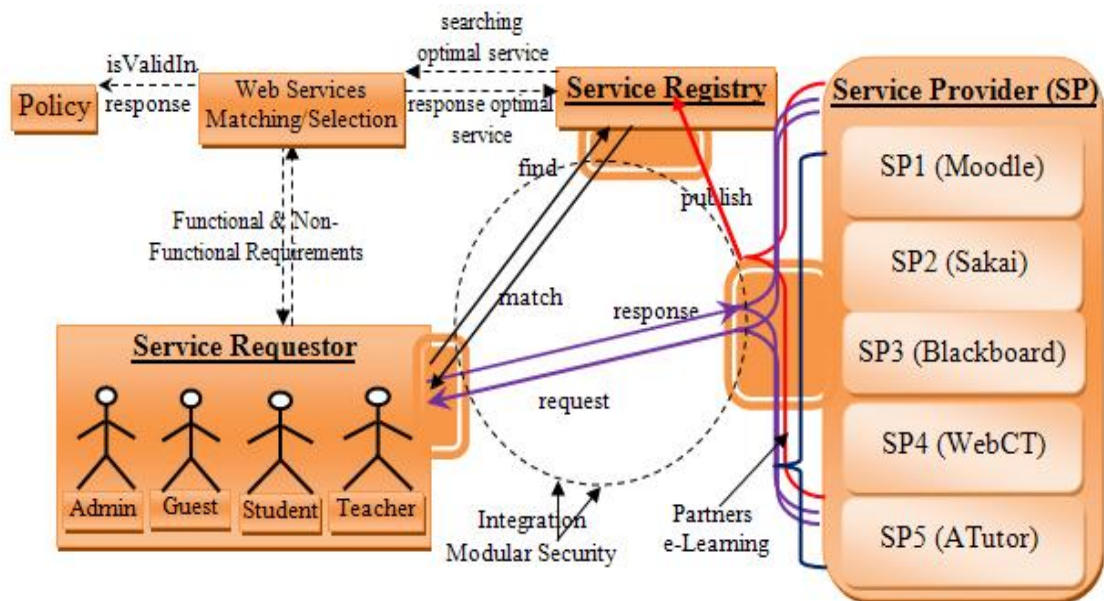


Figure 10.1: The Security Environment of a Collaborative E-learning Model using a Web services technology

As we can see in Figure 10.1, the service provider for all partners, service registry and service requestor are control by integration modular security. This system will provide a cooperative and interactive VLE environment for students to interact with teachers or intelligent agents in terms of their preferences and interests from any universities at any time by using mobile or private access tools. Also, it will enable all teachers in partners e-learning to work together and share material by connecting individual courses together, which are hosted in different VLEs. However, there are some requirements, especially security and privacy issues, and these are :

10.1.1 Security and Privacy

Exchange information over the Internet is vital, because it is an insecure and untrustable public network infrastructure, prone to malicious attacks by professional and amateur intruders. Therefore, security has become a critical issue in e-learning over recent years, especially for large institutions. These institutions should use Secure Sockets Layer (SSL) and a login system to protect their applications and databases. Security should be handled in layers, and the applications, the networks and data need to be protected by using XML security, which is one of the better methods for protecting data [171].

Neither the data structure nor XML itself are sensitive, but they are vulnerable if a hacker gains access to an application and begins sending their own XML instruction sets. The XML must be signed on the sending side and verified for both integrity and sender authenticity by providing additional authentication of the commands and digital signatures using XML signatures. By adding a layer of complexity to the application, the system in turn adds an extra layer of security as in a Collaborative E-learning Model in Figure 10.1 [171].

Privacy is also an important issue, and can be defined as the control that individuals have over their personal space, free from interference by other users and

institutions. The information held in VLE products is of concern to students as the institutions handle and control data about them.[166].

Our future work will present and implement a policy-based privacy and security management scheme for collaborative VLE products. Privacy and security management is achievable in e-learning products by using Ponder and related techniques, as policy language specification.

10.1.2 Trust

Trust is a confident reliance. We may have confidence in people, events, or at least in our beliefs, but if we do not rely on them, our confidence alone does not amount to trust. Trust is a specification that uses the secure messaging mechanisms of Web services security to facilitate trust relationships in diverse Web service environments. However, each party needs to determine for themselves if they can trust the assertions of the other party. The Web Services Trust (WS-Trust) language uses the secure messaging mechanisms of WS-Security to define additional primitives and extensions for issuing, exchanging, and validating security tokens. Using these extensions, applications can engage in secure communications while using the basic SOAP+WSDL+UDDI+HTTP framework for Web services [4].

10.1.3 Flexibility

Flexibility has become one of the more important strategic competitive tools, and refers to the availability of alternative resources. These resources may have diverse parameters, especially these related to the physical and operating systems. Typically, flexibility is one of the major advantages of using Web services, which are a set of protocols based on XML/SOAP over HTTP. The service is accessible using SOAP, which offers more flexibility to developers because they can use the technology that they are most comfortable using [139].

References

- [1] M. P. McCahill and J. Lombardi, “Design for an extensible croquet-based framework to deliver a persistent, unified, massively multi-user, and self-organizing virtual environment,” in *C5 '04: Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing*, (Washington, DC, USA), pp. 71–77, IEEE Computer Society, 2004.
- [2] C. chao Su, “An open source portal for educators,” in *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT05)*, vol. 9 of 0-7695-2338, pp. 961 – 962, 2005.
- [3] S. Wilson. Future of VLE - The Visual Version, Scott Wilson’s Website, 2005. retrieved April 15, 2007 from <http://www.cetis.ac.uk/members/scott/blogview?entry=20050125170206>.
- [4] M. Huhns and M. Singh, *Service-Oriented Computing: Semantics, Processes, Agents*. 0470091487, England: John Wiley and Sons Ltd, 2005.
- [5] D. Fensel and C. Bussler. The Web Service Modeling Framework WSMF, Vrije Universiteit Amsterdam, Faculty of Sciences, Division of Mathematics and Computer Science, 2002. De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands.
- [6] M. Alter, “Web services.” University of Technology, Graz, Austria, February 2005. 9730751.

- [7] C. COMERT. Web Services and National Spatial Data Infrastructure (NSDI), KTU, Geodesy and Photogrammetry Department, 6108 Trabzon, Turkey, 2004. Commission IV, WG IV/4 Spatial Data Infrastructures, TS.
- [8] J. Fan and S. Kambhampati, “A snapshot of public web services,” *SIGMOD Rec.*, vol. 34, no. 1, pp. 24–32, 2005.
- [9] N. Mukhi, R. Konuru, and F. Curbera, “Cooperative middleware specialization for service oriented architectures,” in *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, (New York, NY, USA), pp. 206–215, ACM Press, 2004.
- [10] A. Sinha and A. Paradkar, “Model-based functional conformance testing of web services operating on persistent data,” in *TAV-WEB '06: Proceedings of the 2006 workshop on Testing, analysis, and verification of web services and applications*, (New York, NY, USA), pp. 17–22, ACM Press, 2006.
- [11] L. Ardissono, A. Goy, and G. Petrone, “Enabling conversations with web services,” in *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, (New York, NY, USA), pp. 819–826, ACM Press, 2003.
- [12] E. Cavanaugh, “Web services: Benefits, challenges, and a unique, visual development solution, product marketing manager,” *Altova WhitePaper*, vol. Suite 314-T 1 Beverly, MA, no. ISBN: 01915-6181, USA, 2006.
- [13] F. Goethals, “New directions in application integration: Web services,” 2002.
- [14] E. Newcomer, *Understanding Web Services XML/ WSOL/ SOAP and UDDI*. Addison-Wesley Professional, May 13 2002. ISBN-10: 0201750813 ISBN-13: 978-0201750812.

-
- [15] J. Nandigam and V. N. Gudivada, “Wsexp: A tool for experimenting with web services,” *WSEXP: A Tool for Experimenting with Web Services*, *J. Comput. Small Coll.*, vol. 22, no. 1, pp. 36–45, 2006.
- [16] J. Yang, “Web service componentization,” *Commun. ACM*, vol. 46, no. 10, pp. 35–40, 2003.
- [17] M. Papazoglou, “Service-oriented computing: concepts, characteristics and directions,” in *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*, pp. 3 – 12, IEEE, 10-12 Dec. 2003. Digital Object Identifier 10.1109/WISE.2003.1254461.
- [18] M. Papazoglou and D. Georgakopoulos, “Introduction,” *Commun. ACM*, vol. 46, no. 10, pp. 24–28, 2003.
- [19] M. Mecella, M. Ouzzani, F. Paci, and E. Bertino, “Access control enforcement for conversation-based web services,” in *WWW ’06: Proceedings of the 15th international conference on World Wide Web*, (New York, NY, USA), pp. 257–266, ACM Press, 2006.
- [20] N. Mendon and J. Silva, “An empirical evaluation of client-side server selection policies for accessing replicated web services,” in *SAC ’05: Proceedings of the 2005 ACM symposium on Applied computing*, (New York, NY, USA), pp. 1704–1708, ACM, 2005.
- [21] S. Yang, J. Hsieh, B. Lan, and J.-Y. Chung, “Composition and evaluation of trustworthy web services,” in *BSN ’05: Proceedings of the IEEE EEE05 international workshop on Business services networks*, (Piscataway, NJ, USA), pp. 5–5, IEEE Press, 2005.
- [22] R. Darbhamulla and P. Tadeipalli, “Web services demystified,” *J. Comput. Small Coll.*, vol. 21, no. 5, pp. 1–2, 2006.

- [23] W.Tsai, Y. Chen, G. Bitter, and D. Miron, “Introduction to service-oriented computing.” Computer Science and Engineering Department, Arizona State University, 2005. retrieved December 05, 2006.
- [24] E. Ambrosi, M. Bianchi, and G. Felici, “When is it convenient to predict the web services completion time?,” in *PDCN’06: Proceedings of the 24th IASTED international conference on Parallel and distributed computing and networks*, (Anaheim, CA, USA), pp. 250–255, ACTA Press, 2006.
- [25] M. Huhns and M. Singh, “Service-oriented computing: Key concepts and principles,” *IEEE Internet Computing*, vol. 9, no. 1, pp. 75–81, 2005.
- [26] U. Srivastava, K. Munagala, J. Widom, and R. Motwani, “Query optimization over web services,” in *in Proc. of the 32nd international conference on Very large data bases*, pp. 355–366, VLDB Endowment, 2006.
- [27] J. Gortmaker, M. Janssen, and R. Wagenaar, “The advantages of web service orchestration in perspective,” in *ICEC ’04: Proceedings of the 6th international conference on Electronic commerce*, (New York, NY, USA), pp. 506–515, ACM Press, 2004.
- [28] N. S. Katia Sycara, Massimo Paolucci and M. Burstein, “Combining services and semantics on the web.” (Software Agents Lab, Carnegie Mellon University) and (Human-Centered Systems Group, BBN Technologies, Cambridge, MA), 2004.
- [29] F. Coyle, ****XML, WEB SERVICES, AND THE DATA REVOLUTION.**** No. chapter 5, Addison-Wesley ,Pearson Education, Inc, 2002.
- [30] D. Austin, A. Barbir, C. Ferris, and S. Garg, “Web services architecture requirements.” W3C, 2004.

- [31] M. Tian, T. Voigt, T. Naumowicz, H. Ritter, and J. Schiller, “Performance considerations for mobile web services,” *Elsevier Computer Communications Journal*, vol. 27, pp. 1097–1105, 2004.
- [32] B. Sleeper and S. Partner, “Defining web services.” Stencil Group, 2001. retrieved October 21, 2006 from <http://www.site.uottawa.ca>.
- [33] T. Kaizu, T. Noro, and T. Tokuda, “A state propagation method for consistency checking of web service function invocations in web applications,” in *ICWE '06: Workshop proceedings of the sixth international conference on Web engineering*, (New York, NY, USA), p. 18, ACM Press, July 10 - 14 2006.
- [34] F. Jammes, A. Mensch, and H. Smit, “Service-oriented device communications using the devices profile for web services,” in *MPAC '05: Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing*, (New York, NY, USA), pp. 1–8, ACM Press, 2005.
- [35] L. Wilkes, “Roi - the costs and benefits of web services and service oriented architecture.” IBM, 2006.
- [36] L. Ardissono, L. Console, A. Goy, G. Petrone, C. Picardi, M. Segnan, and D. T. Dupr, “Advanced fault analysis in web service composition,” in *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, (New York, NY, USA), pp. 1090–1091, ACM Press, 2005.
- [37] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, “Unraveling the web services web: An introduction to soap, wsdl, and uddi,” *Internet Computing, IEEE*, vol. 6, pp. 86 – 93, 2002.
- [38] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. 3540440089, Springer-Verlag Berlin and Hei-

- delberg GmbH & Co. K, March 2003. ISBN-10: 3540440089 ISBN-13: 978-3540440086.
- [39] S. Tai, R. Khalaf, and T. Mikalsen, “Composition of coordinated web services,” in *in Proc. of the 5th ACM/IFIP/USENIX international conference on Middleware*, (New York, NY, USA), pp. 294–310, Springer-Verlag New York, Inc., 2004.
- [40] V. Vasudevan, “Web services primer.” O’Reilly Media, Inc., 2001.
- [41] H. K. K. Gottschalk, S. Graham and J. Shell, “Introduction to web services architecture,” *IBM Systems journal*, vol. 41, no. 2, pp. 170–177, 2002.
- [42] A. Tsalgatidou and T. Pilioura, “An overview of standards and related technology in web services,” *Distrib. Parallel Databases*, vol. 12, no. 2-3, pp. 135–162, 2002.
- [43] J. Myerson, “Advancing the web services stacks,” *A proposed synthesis of IBM’s Web services architecture stack and new IBM technologies*, Jun 2002. Systems Architect/Engineer.
- [44] D. Distant, G. Canfora, S. Tilley, and S. Huang, “Redesigning legacy applications for the web with uwat+: a case study,” in *ICSE ’06: Proceeding of the 28th international conference on Software engineering*, (New York, NY, USA), pp. 482–491, ACM Press, 2006.
- [45] D. P. Robert C. Seacord and G. A. Lewis, *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices (Paperback)*. Addison-Wesley Professional, 2003.
- [46] M. Brodie and M. Stonebraker, *Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach*. 1558603301, USA: Morgan Kaufmann Publishers, Inc., March 1995. 978-1558603301.

- [47] S. Software, “Web services.” retrieved May 7, 2007 from <http://www.seagullsoftware.com/solutions/technology/webservices.html>topofpage, 2006.
- [48] F. Silva and P. Rosa, “The quest for the web services stack: a fast trip,” in *in Proc. of the 6th international conference on Web engineering*, (New York, NY, USA), pp. 93–94, ACM Press, 2006.
- [49] M. Fernandez, N. Onose, and J. Simeon, “Yoo-hoo!: building a presence service with xquery and wsdl,” in *in Proc. of the 2004 ACM SIGMOD international conference on Management of data*, (New York, NY, USA), pp. 911–912, ACM Press, 2004.
- [50] S. Hollenbeck, M. Rose, and L. Masinter, “Guidelines for the use of extensible markup language (xml) within ietf protocols,” *RFC Editor*, January 2003.
- [51] T. Takase, H. MIYASHITA, T. Suzumura, and M. Tatsubori, “An adaptive, fast, and safe xml parser based on byte sequences memorization,” in *WWW ’05: Proceedings of the 14th international conference on World Wide Web*, (New York, NY, USA), pp. 692–701, ACM Press, 2005.
- [52] A. Nghiem, *IT Web Services: IT Web Services: A Roadmap for the Enterprise*, vol. Chapter 2 the Basic Web Services Stack of 0130097195. Prentice Hall PTR, Har/Cdr edition, October 2002.
- [53] V. Turau, “Httpexplorer: exploring the hypertext transfer protocol,” in *in Proc. of the 8th annual conference on Innovation and technology in computer science education*, (New York, NY, USA), pp. 198–201, ACM Press, 2003.
- [54] J. Klensin, “Simple mail transfer protocol.” RFC Editor, Network Working Group, April 2001. United States.

- [55] J. Jung and E. Sit, “An empirical study of spam traffic and the use of dns black lists,” in *in Proc. of the 4th ACM SIGCOMM conference on Internet measurement*, (New York, NY, USA), pp. 370–375, ACM Press, 2004.
- [56] B. Curtin, “Internationalization of the file transfer protocol,” *Network Working Group, The Internet Society*, 1999.
- [57] M. Rose, “The blocks extensible exchange protocol core,” *RFC Editor*, vol. United States, 2001.
- [58] P. Louridas, “Soap and web services,” *EEE Software, IEEE Computer Society Press*, 2006.
- [59] J. Fuller, H. Fuecks, K. Egervari, B. Waters, D. Solin, J. Stephens, and L. Reynolds, *Professional PHP Web Services*. 1861008074, Wrox Press, 1st edition ed., 2003. 13: 978-1861008077.
- [60] R. Richards., *Pro PHP XML and Web Services*. 1590596331, Apress, March 27 2006. 13: 978-1590596333.
- [61] D. Technologies, “Jdbc definitions,” 2006. retrieved April 22, 2007.
- [62] M. A. Rahaman, A. Schaad, and M. Rits, “Towards secure soap message exchange in a soa,” in *SWS ’06: Proceedings of the 3rd ACM workshop on Secure web services*, (New York, NY, USA), pp. 77–84, ACM Press, 2006.
- [63] N. Apte, K. Deutsch, and R. Jain, “Wireless soap: optimizations for mobile wireless web services,” in *WWW ’05: Special interest tracks and posters of the 14th international conference on World Wide Web*, (New York, NY, USA), pp. 1178–1179, ACM Press, 2005.
- [64] L. Lee, “Xml-rpc.” XML-RPC.com, 2003. retrieved April 22, 2007.

- [65] J. Johnston, E. Dumbill, S. Laurent, and J. Posner, *Programming Web Services with XML-RPC*. 0596001193, O'Reilly Media, 1 edition ed., June 2001. 13: 978-0596001193.
- [66] P. Sarang, C. Browne, D. Ayala, and V. Chopra., *Professional Open Source Web Services*. 1861007469, Wrox Press, July 2002. 13: 978-1861007469.
- [67] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web services description language (wsdl) 1.1." Ariba, International Business Machines Corporation Microsoft, 2001.
- [68] V. de Castro, E. Marcos, and B. Vela, "Representing wsdl with extended uml," 2003. retrieved Nov 5, 2007 from http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r51_art1_c.pdf.
- [69] A. Paar and W. Tichy, "Programming language inherent support for constrained xml schema definition data types and owl dl," in *in Proc. of IEEE Computer Society*, 2006.
- [70] Y. Shiran, "Summary table of wsdl elements," 2001. Ph.D, retrieved February 12, 2007.
- [71] S. Banerjee, S. Basu, S. Garg, S. Garg, S.-J. Lee, P. Mullan, and P. Sharma, "Scalable grid service discovery based on uddi," in *MGC '05: Proceedings of the 3rd international workshop on Middleware for grid computing*, (New York, NY, USA), pp. 1–6, ACM, 2005.
- [72] C. Boulton, "W3c:ws-addressing is good to go," May 2006.
- [73] W. Iverson, "Web services in action: Integrating with the ebay marketplace," 2004.
- [74] A. Skonnard, "The birth of web services." Microsoft Corporation, 2005.

-
- [75] L. Esposito, “An overview of the global xml web services architecture platform,” 2003.
- [76] G. Gianforte, “The insiders guide to next-generation web customer service.” retrieved December 11, 2006.
- [77] A. Toth and P. Toth, “Virtual learning aspects of curriculum development in technical teacher training,” in *INES '06. Proceedings. International Conference on Intelligent Engineering Systems*, (London), pp. 308–313, 2006.
- [78] O. Conlan, *Multi-Model: Metadata Driven Approach to Personalised e-Learning Services*. Doctor of philosophy, University of Dublin, Trinity College, Department of Computer Science, Trinity College, , Dublin, 2005.
- [79] A. Chavan and S. Pavri, “Open-source learning management with moodle,” *linuxjournal*, p. 7817, December 2004. Specialized Systems Consultants.
- [80] A. Dong and H. Li, “Ontology-based information integration in virtual learning environment,” in *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, 0-7695-2415-X, (Dept. of Comput. Sci., North Dakota State Univ., Fargo, ND, USA;), pp. 762– 765, IEEE, 19-22 Sept 2005.
- [81] S. Graham, S. Simeonov, T. Boubez, G. D. D. Davis, Y. Nakamura, and R. Neyama, *Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI*. 0672321815, Pearson Education, 1st ed., December 2001.
- [82] A. Dong and H. Li, “Multimedia access platform for virtual learning environment,” in *IEEE International Conference on Electro Information Technology*, 0-7803-9232-9, (North Dakota State Univ., Fargo, ND, USA), pp. 6 pp.–, IEEE, May 2005.
- [83] M. Dougiamas, “Moodle.” retrieved April 19, 2007 from www.MOODLE.org, 2008.

- [84] S. Dunn, "A project for city university centre for hci design." retrieved April 13, 2007 from <http://www.saradunn.net/VLEproject/>, April 2003.
- [85] M. Berry, "An investigation of the effectiveness of moodle in primary education." Deputy Head, St Ives School, Haslemere, 2005. retrieved April 5, 2007 from <http://Moodlemoot.org/mod/resource/view.php?id=19>.
- [86] J. Massy, "Study of the e-learning suppliers market in europe," Master's thesis, Danish Technological Institute, Heriot-Watt University, 2004.
- [87] C. Cheng and J. Yen, "Virtual learning environment (vle): a web-based collaborative learning system," in *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, vol. 1 of 0-8186-8255-8, (Kohala Coast, HI, USA), pp. 480–491, Univ. of Hong Kong, IEEE, 6-9 Jan 1998.
- [88] A. Kumar, R. Pakala, R. Ragade, and J. Wong, "The virtual learning environment system," in *Frontiers in Education Conference, 1998. FIE '98. 28th Annual*, vol. 2 of 0-7803-4762-5, (Tempe, AZ, USA), pp. 711–716, Louisville Univ, IEEE, 4-7 Nov 1998.
- [89] J. Bruce and N. Curson, "Uea virtual learning environment." UEA, Product evaluation report, Learning Technology Group, 2001. retrieved April 19, 2007 from <http://www.uea.ac.uk/ltg/blackboard/VLEreport.pdf>.
- [90] S. Britain and O. Liber, "A framework for pedagogical evaluation of virtual learning environments." JTAP, JISC Technology Applications Programme, University of Wales-Bangor, 1999. retrieved April 9, 2007 from <http://www.leeds.ac.uk/educol/documents/00001237.htm>.
- [91] J. McClelland, "Evolving web-based delivery: managing the transition from vle's designed by academics to commercial vle's, the post-graduate experi-

- ence,” in *Proceedings on 28th in Euromicro Conference*, 1089-6503, (Liverpool), pp. 154–159, IEEE, 2002.
- [92] M. Schulte-Mecklenbeck, “Blending old teaching methods with a cutting-edge virtual learning environment,” *Brave New World ... Wide Web*, vol. 17, October 2004.
- [93] R. O’Leary and A. Ramsden, “Virtual learning environments.” University of Bristol, 2002. The Handbook for Economics Lecturers.
- [94] A. Marshall, H. Noble, and S. Lee, “Value added moodle: Experimental integration with the bodington vle.” http://bodington.org/art_moodle.pdf, 2005. Learning Technologies Group, Oxford University Computing Services,.
- [95] J. Gray and R. Everett, “Introducing managed learning environments (mles).” JISC Interoperability Project, 2000. retrieved April 9, 2007 from <http://www.jisc.ac.uk>.
- [96] J. Lombardi and M. P. McCahill, “Enabling social dimensions of learning through a persistent, unified, massively multi-user, and self-organizing virtual environment,” in *C5 ’04: Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing*, (Washington, DC, USA), pp. 166–172, IEEE Computer Society, 2004.
- [97] S. Graf and B. List, “An evaluation of open source e-learning platforms stressing adaptation issues,” in *ICALT ’05: Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies*, (Washington, DC, USA), pp. 163–165, IEEE Computer Society, 2005.
- [98] M. Dougiamas and P. Taylor, “Interpretive analysis of an internet-based course constructed using a new courseware tool called moodle,” in *HERDSA 2002 conference*, (Perth, Australia), Curtin University of Technology, 20.

- [99] Y. Perrie, "Virtual learning environment," *The Pharmaceutical Journal*, vol. 270, pp. 794–795, 2003.
- [100] K. Moyle, "Open source software and australian school education." education.au limited, MCEETYA ICT in Schools Taskforce, August 2003. retrieved August 27, 2007 from <http://www.educationau.edu.au/jahia/webdav/site/myjahiasite/shared/papers/>.
- [101] K. Coar, "The criterias of open source software." Opensource.org site, 2006. retrieved July 03, 2007 from <http://opensource.org>.
- [102] C. Smart, "Choosing open source solutions." JISC e-Framework for Education and Research, published on 17 May 2005. retrieved March 23, 2007 from <http://www.elearning.ac.uk/features/oss>.
- [103] E. McMahon, "Future use and development of the 'learnwise' virtual learning environment." BECTA, 2001. retrieved April 15, 2007 from <http://ferl.becta.org.uk/display.cfm?page=13&resID=1598>.
- [104] U. Nation, "Online higher education-issues for developing countries united nation conference on trade and development." E-commerce and development report, 2004. New York and Geneva.
- [105] M. Weller, "Vle 2.0 and future directions in learning environments," in *Proceedings of The First International LAMS Conference 2006: Designing the Future of Learning*, (Institute of Educational Technology, Milton Keynes, UK), pp. 99–106, The Open University, 2006.
- [106] J. Malins and I. Pirie, "Developing a virtual learning environment for art & design: A constructivist approach." Grays School of Art, The Robert Gordon University, Aberdeen, 2002.

- [107] S. Yuuichi, K. Toshihiro, Y. Seisuke, and N. Hiroshi, “Web-based rapid authoring tool for lms quiz creation,” in *ITHET '06. 7th International Conference on Information Technology Based Higher Education and Training, 2006.*, 1-4244-0406-1, (Ultimo, Australia), pp. 617–620, Kumamoto University, IEEE, 2006.
- [108] G. Tortora, M. Sebillo, G. Vitiello, and P. D’Ambrosio, “A multilevel learning management system,” in *SEKE '02: Proceedings of the 14th international conference on Software engineering and knowledge engineering*, (New York, NY, USA), pp. 541–547, ACM, 2002.
- [109] M. Zenha-Rela and R. Carvalho, “Work in progress: Self evaluation through monitored peer review using the moodle platform,” in *Frontiers in Education Conference, 36th Annual*, 0190-5848, (San Diego, CA), pp. 26–27, Dept. Eng. Informatica, Coimbra Univ, IEEE, 2006.
- [110] K. Brandl, “Are you ready to ”moodle”?,” *Language Learning & Technology*, vol. 9, pp. 16–23, May 2005. University of Washington.
- [111] C. Martin, N. Morris-Cotterill, and M. Smith, “Open source software for the education market.” eduforge, March 2004. retrieved November 5, 2006 from <http://eduforge.org>.
- [112] EduTools, “Course management systems.” EduTools, 2006. retrieved July 19, 2006 from <http://www.edutools.info/static.jsp?pj8&pageHOME>.
- [113] L. Jolla and H. Motulsky, “Graphpad prism.” GraphPad Software, Inc., 2008. retrieved June 29, 2008 from <http://www.graphpad.com/prism/Prism.htm>.
- [114] H. Wharekura-tini and kahautu Aotearoa, “Technical evaluation of selected learning management systems,” Master’s thesis, Catalyst IT Limited, Open Polytechnic of New Zealand, 2004.

- [115] N. Sclater, “Moodle : Transforming learning transforming institutions,” in *MOODLE Regional User Group Conference*, 1904811299, The Open University, Packt Publishing, 17th March 2006.
- [116] V. Alvarez, “e-learning survey.” Department of Computer Science at the University of Oviedo, 2008. retrieved April 9, 2007 from <http://www.di.uniovi.es/victoralvarez/survey/>.
- [117] J. Cole, *Using MOODLE: Teaching with the Popular Open Source Course Management System*. 0596008635, O’Reilly, 2005.
- [118] M. Dougiamas, “Moodle:a virtual learning environment for the rest of us,,” *TESL-EJ*, vol. 8, pp. 1–8, September 2004.
- [119] B. Williams and M. Dougiamas., “Moodle for teachers.” Trainers and Administrators of remote-Learner.net, Creative Commons Noncommercial copyright, MOODLE.org community,, January 2005.
- [120] K. Koh, “Moodle as a course management system.” [eslweb.org](http://www.eslweb.org), June 2006. retrieved June 15, 2006 from <http://www.eslweb.org/criticalreviews/MOODLEm.pdf>.
- [121] J. Itmazi, *Flexible Learning Management System To Support Learning In The Traditional And Open Universities*. Phd program: Advanced methods & technique of software development, Software Engineering Department, Computer Engineering School, Granada University, Granada, Spain, 2005.
- [122] S. Shearer, “Open source software in education.” The Compton School, 2003. London.
- [123] J. MacKenzie, A. Muirhead, and S. Mann, “The use and usefulness of moodle.” University of Clagow, 2006. retrieved July 15, 2006 from <http://www.gla.ac.uk/faculties/education/vle/docs/useandusefulness.ppt>.

-
- [124] M. Feldstein, “British columbia distance learning program recommending moodle.” British Columbia Distance Learning Program, April 2006. retrieved June 19, 2006 from <http://mfeldstein.com/index.php/weblog/2006/04/>.
- [125] S. L. Adam Marshall, Howard Noble, “Value added moodle: Experimental integration with the bodington vle.” Learning Technologies Group, Oxford University Computing Services, 2005. <http://bodington.org/artmoodle.pdf>.
- [126] B. Bos, K. Munoz, and J. Duzer, “Blackboard vs. moodle : A comparison of satisfaction with online teaching and learning tools.” retrieved June 19, 2006 from [http://www.humboldt.edu/jdv1/Moodle /all.htm](http://www.humboldt.edu/jdv1/Moodle/all.htm), 2005.
- [127] S. Ambler, *The Object Primer: Agile Model Driven Development with UML 2*. 0-521-54018-6, <http://www.ambyssoft.com/books/theObjectPrimer.html>: Cambridge University Press, 3 ed., 2004.
- [128] S. Parsa and M. Damanafshan, “Seamless secure development of systems: From modeling to enforcement of access control policies,” in *AICCSA 07. IEEE/ACS International Conference on Computer Systems and Applications, 2007.*, 1-4244-1031-2, (Amman), pp. 799–806, IEEE, 2007.
- [129] V. Garousi, L. Briand, and Y. Labiche, “Traffic-aware stress testing of distributed real-time systems based on uml models using genetic algorithms,” *J. Syst. Softw.*, vol. 81, no. 2, pp. 161–185, 2008.
- [130] A. Orlovsky and D. Raz, “Decentralized enforcement of security policies for distributed computational systems,” in *SAC ’07: Proceedings of the 2007 ACM symposium on Applied computing*, (New York, NY, USA), pp. 241–248, ACM, 2007.
- [131] IBM, “Concepts and planning.” IBM Corp, 2005. 6 ed. TXSeries for Multi-platforms.

- [132] S. Singh, S. Xu, and H. Kaur, “Efficient development of highly reusable distributed systems using the tcao,” in *ACST’07: Proceedings of the third conference on IASTED International Conference*, (Anaheim, CA, USA), pp. 455–464, ACTA Press, 2007.
- [133] A. Takahashi and T. Kinoshita, “A design and operation model for agent-based flexible distributed system,” in *IAT ’06: Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, (Washington, DC, USA), pp. 88–95, IEEE Computer Society, 2006.
- [134] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*. 0471389226, Wiley, 1 ed., 2001.
- [135] I. Foster, K. Czajkowski, D. Ferguson, J. Frey, S. Graham, T. Maguire, D. Snelling, and S. Tuecke, “Modeling and managing state in distributed systems: The role of oksi and wsrf,” *Proceedings of the IEEE*, vol. 93, no. 3, pp. 604–612, 2005.
- [136] Z. Singhera and A. Shah, “Extended web services framework to meet non-functional requirements,” in *ICWE ’06: Workshop proceedings of the sixth international conference on Web engineering*, (New York, NY, USA), p. 21, ACM, 2006.
- [137] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, “The ponder policy specification language,” in *POLICY ’01: Proceedings of the International Workshop on Policies for Distributed Systems and Networks*, (London, UK), pp. 18–38, Springer-Verlag, 2001.
- [138] P. Reinheimer, *Professional Web APIs with PHP: EBay, Google, Paypal, Amazon, FedEx Plus Web Feeds (Programmer to Programmer)*. 978-0764589546, John Wiley & Sons, 1 ed., 2006.

- [139] F. Chan, R. Bhagwat, and S. Wadhwa, “Flexibility performance: Taguchi’s method study of physical system and operating control parameters of fms,” *Robot. Comput.-Integr. Manuf.*, vol. 23, no. 1, pp. 25–37, 2007.
- [140] K. Lee, J. Jeon, W. Lee, S.-H. Jeong, and S.-W. Park, “Qos for web services: Requirements and possible approaches.” W3C Working Group, 2003. retrieved March 23, 2007 from <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>.
- [141] E. Arisholm and D. Sjoberg, “Towards a framework for empirical assessment of changeability decay,” *Journal of Systems and Software*, vol. 53, no. 1, pp. 3–14, 2000.
- [142] L. Baresi, R. Heckel, S. Thne, and D. Varr, “Modeling and validation of service-oriented architectures: Application vs. style,” in *ESEC/FSE-11: Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*, (New York, NY, USA), pp. 68–77, ACM, 2003.
- [143] L. Baresi, R. Heckel, S. Thne, and D. Varr, “A generic, platform-independent architectural style.” Graduate School Dynamic Intelligent Systems, University of Paderborn, D-33095 Paderborn, Germany, September 2003. retrieved June 5, 2007 from www.upb.de/cs/ag-engels/ag_engl/People/Thoene/MRDSA/.
- [144] L. Baresi, R. Heckel, S. Thne, and D. Varr, “A uml-profile for service-oriented architectures.” Graduate School Dynamic Intelligent Systems, University of Paderborn, D-33095 Paderborn, Germany, September 2003. retrieved June 7, 2007 from www.upb.de/cs/ag-engels/ag_engl/People/Thoene/MRDSA/.
- [145] M. Chonoles and J. Schardt, *UML 2 for Dummies*. 0764526146, <http://www.amazon.com/UML-Dummies-Michael-Jesse-Chonoles/dp/0764526146>: For Dummies, July 2003.

- [146] D. Braun, J. Sivils, A. Shapiro, and J. Versteegh, “Unified modeling language (uml) tutorial.” Object Oriented Analysis and Design Team, Spring 2001. retrieved April 19, 2007 from http://atlas.kennesaw.edu/dbraun/csis4650/A&D/UML_tutorial/index.htm.
- [147] K. Falkovych, M. Sabou, and H. Stuckenschmidt, “Uml for the semantic web: Transformation-based approaches,” 2003.
- [148] G. Tegos, D. Stoyanova, and K. Onkov, “E-learning of trend modeling in a web-environment,” *SIGCSE Bull.*, vol. 37, no. 2, pp. 70–74, 2005.
- [149] A. Charfi and M. Mezini, “Hybrid web service composition: business processes meet business rules,” in *ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing*, (New York, NY, USA), pp. 30–38, ACM, 2004.
- [150] P. Corporation, “Separation of business logic from presentation logic in web applications,” July 2003. retrieved July 03, 2007 from <http://www.paragoncorporation.com/ArticleDetail.aspx?ArticleID=21>.
- [151] M. Denford, J. Leaney, and T. O’Neill, “Non-functional refinement of computer based systems architecture,” in *ECBS '04: Proceedings of the 11th IEEE International Conference and Workshop on Engineering of Computer-Based Systems*, (Washington, DC, USA), p. 168, IEEE Computer Society, 2004.
- [152] Object-Management-Group, “Uml 2.0 superstructure final adopted specification.” Object Management Group, 2003. retrieved April 9, 2007 from <http://www.omg.org/docs/ptc/03-08-02.pdf>.
- [153] R. Heckel, M. Lohmann, and S. Thone, “Towards a uml profile for service-oriented architectures,” in *on Syntactic and Semantic Integration of Visual*

- Modeling Techniques*, vol. 1, pp. 1–6, the University of Paderborn, Elsevier Science, 2003.
- [154] J. Hausmann, R. Heckel, and M. Lohmann, “Model-based development of web services descriptions enabling a prices matching concept,” *International Journal of Web Services Research*, vol. 2, no. 2, pp. 1–17, 2005.
- [155] A. Al-Ajlan and H. Zedan, “The extension of web services architecture to meet the technical requirements of virtual learning environments (moodle),” in *CCES08-IEEE International Conference on Computer Engineering & Systems*, (Cairo, Egypt), Ain Shams University - Faculty of Engineering, IEEE, November 25-27 2008.
- [156] H. Shayani and P. Bentley, “A more bio-plausible approach to the evolutionary inference of finite state machines,” in *GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, (New York, NY, USA), pp. 2937–2944, ACM, 2007.
- [157] B. Lee and E. Lee, “Interaction of finite state machines and concurrency models,” in *Proceeding of Thirty Second Annual Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, California), November 1998.
- [158] S. H. Rodger, J. Lim, and S. Reading, “Increasing interaction and support in the formal languages and automata theory course,” *SIGCSE Bull.*, vol. 39, no. 3, pp. 58–62, 2007.
- [159] M. Yingling, “Mobile moodle,” *J. Comput. Small Coll.*, vol. 21, no. 6, pp. 280–281, 2006.
- [160] D. Garcia and M. Toledo, “Semantics-enriched qos policies for web service interactions,” in *WebMedia '06: Proceedings of the 12th Brazilian symposium on Multimedia and the web*, (New York, NY, USA), pp. 35–44, ACM, 2006.

- [161] Y. Zhao and F. Parisi-Presicce, “Policy analysis and verification by graph transformation tools,” in *Electronic Notes in Theoretical Computer Science*, (USA), George Mason University, elsevier, 2004. www.elsevier.nl/locate/entcs.
- [162] E. C. Lupu, *A Role-Based Framework for Distributed Systems Management*. PhD thesis, Imperial College of Science Technology and Medicine, Department of Computing, London, UK, 1997.
- [163] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, “Ponder: A language for specifying security and management policies for distributed systems, the language specification, version 2.3,” tech. rep., Imperial College of Science Technology and Medicine, Department of Computing, London, UK., 2001.
- [164] V. C. Hu, E. Martin, J. Hwang, and T. Xie, “Conformance checking of access control policies specified in xacml,” in *COMPSAC ’07: Proceedings of the 31st Annual International Computer Software and Applications Conference - Vol. 2- (COMPSAC 2007)*, (Washington, DC, USA), pp. 275–280, IEEE Computer Society, 2007.
- [165] E. Martin, “Testing and analysis of access control policies,” in *ICSE COMPANION ’07: Companion to the proceedings of the 29th International Conference on Software Engineering*, (Washington, DC, USA), pp. 75–76, IEEE Computer Society, 2007.
- [166] C. Yang, F. Lin, and H. Lin, “Policy-based privacy and security management for collaborative e-education systems,” in *Proceedings of the 5th IASTED International Multi-Conference Computers and Advanced Technology in Education (CATE 2002)*, NRC 44906, (Cancun, Mexico), May 20-22 2002.
- [167] M. A. Azadeh, M. Anvari, and H. Izadbakhsh, “An integrated fdea-pca method as decision making model and computer simulation for system optimization,”

- in *SCSC: Proceedings of the 2007 summer computer simulation conference*, (San Diego, CA, USA), pp. 609–616, Society for Computer Simulation International, 2007.
- [168] AGG, “A brief description of agg.” AGG Website, 01 2008. retrieved April 19, 2008 from <http://tfs.cs.tu-berlin.de/agg/docu.html>.
- [169] P. Lavin, *Object-Oriented PHP: Concepts, Techniques, and Code*. 1593270771, No Starch Press, 2006.
- [170] B. Johnston, “An introduction to web services enabled with php.” NuSphere Corporation, 2002. retrieved July 15, 2006 from www.nusphere.com.
- [171] A. Al-Ajlan and H. Zedan, “Towards the dynamic web services selection of e-learning services,” in *Proceeding of the Saudi Innovation Conference 2007, ICT Workshop*, vol. 1 of 978-0-955104-92-3, (Newcastle, UK), pp. 657–668, Newcastle University, Saudi Clubs in UK and Ireland term 26th., May 2007.
- [172] A. Al-Ajlan and H. Zedan, “E-learning (moodle) based on service oriented architecture,” in *EADTU’s 20th Anniversary Conference 2007*, vol. 1 of 978-0-8923-86-3, (Lisbon, Portugal), pp. 62–70, Universidade Aberta, EADTU, 8-9 November 2007.
- [173] D. Thomas, “The pear::soap implementation.” Evolt.org, 2002. retrieved April 15, 2007 from http://www.evolt.org/article/The_PEAR_SOAP_Implementation/21/49993/.
- [174] H. Yang, “Php soap extension.” PHP Tutorials - Herong’s Tutorial Notes, 2006. retrieved June 29, 2008 from <http://www.herongyang.com/php/php-soap.html>.
- [175] s. snichol, “Nusoap - soap toolkit for php.” SourceForge.net, 2005. retrieved June 29, 2008 from <http://www.herongyang.com/php/php-soap.html>.

- [176] N. Dulay, E. Lupu, M. Sloman, and N. Damianou, “A policy deployment model for the ponder language,” in *Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on Integrated Network Management*, 0-7803-6719-7, pp. 529–543, Imperial Coll. of Sci., Technol. & Med., London, IEEE, 2001.
- [177] T. Connolly and C. Begg, *Database Systems: A Practical Approach to Design, Implementation and Management*. 0321210255, Addison Wesley, 4 edition ed., 24 May 2002.
- [178] E. Schmitter, *Modelling, analysis and evaluation of systems architectures*, vol. 1105/1996 of 978-3-540-61478-4, book chapter Computer Aided Systems Theory CAST ’94, pp. 240–251. Springer Berlin / Heidelberg, 2006.
- [179] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard, “Web services architecture.” W3C Working Group, February 2004. retrieved July 29, 2008 from <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>.
- [180] A. Al-Ajlan and H. Zedan, “Why moodle,” in *Proc. 12IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, (Kunming, China), IEEE Press, 2008.

Appendices

Appendix A

Publications

During the course of the incremental research, the results have been reported in a number of scientific papers.

A.1 Journals

1. Y. Al-Saawy, A. Al-Ajlan, K. Al-Drewiesh and A. Bajahzer, ‘An Efficient Search Agent Software for Multi-Agent Systems Using Finite State Machine Technology ’ *International Journal of Digital Content Technology and its Application (JDCTA)*, vol. 3(1), pp. 73-87, Korea, March-2009.

A.2 Conferences

1. A. Al-Ajlan, ‘The Evolution of Open Source Software using Eclipse Metrics’, *in Proc. of NISS09-IEEE International Conference on New Trends in Information and Service Sciences (NISS2009)*, IEEE Computer Society, Beijing, China, June-2009.
2. Y. Al-Saawy, A. Al-Ajlan, K. Al-Drewiesh and A. Bajahzer, ‘The Development of Multi-Agent System using Finite State Machine’, *in Proc. of NISS09-IEEE*

International Conference on New Trends in Information and Service Sciences (NISS2009), IEEE Computer Society, Beijing, China, June-2009.

3. F. Lahwal, M. Amaimin, and A. Al-Ajlan, 'Framework of Dynamic E-learning Environment for Interactive Multimedia of Media Programming', in *Proc. of NISS09-IEEE International Conference on New Trends in Information and Service Sciences (NISS2009)*, IEEE Computer Society, Beijing, China, June-2009.
4. A. Al-Ajlan, 'An Efficient Approach to measure the Open Source Software Evolution', in *Proc. of The 3rd Saudi International Conference (SIC-2009)*, ICT Workshop, The University of Surrey, Saudi Clubs in UK and Ireland term 29th, Guildford, UK, June-2009.
5. A. Al-Ajlan and H. Zedan, 'The Extension of Web Services Architecture to Meet the Technical Requirements of Virtual Learning Environments (Moodle)', in *Proc. of ICCES08-IEEE International Conference on Computer Engineering & Systems*, IEEE Computer Society, Cairo, Egypt, Nov-2008.
6. A. Bajahzar, A. Al-Ajlan and H. Zedan, 'Exercise Services for E-learning in Higher Education with Open Source Software (Moodle)', in *Proc. of The IADIS International Conference Www/Internet 2007*, IADIS Press, Freiburg, Germany, Oct-2008.
7. A. Al-Ajlan and H. Zedan, 'Why Moodle', in *Proc. 12IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, IEEE Computer Society, Kunming, China, Oct-2008.
8. A. Al-Ajlan and H. Zedan, 'A Comparison between Moodle and other Virtual Learning Environment Platforms', in *Proc. of the Saudi Innovation Conference 2008*, ICT Workshop, Leeds University, Saudi Clubs in UK and Ireland term 27th, Leeds, UK, Jun-2008.

9. A. Al-Ajlan and H. Zedan, 'E-learning (MOODLE) Based on Service Oriented Architecture', in *Proc. of the EADTU's 20th Anniversary Conference 2007*, Lisbon, Portugal, Nov-2007.
10. A. Al-Ajlan and H. Zedan, 'The Web Services Selection of Virtual Learning Environment Services', in *Proc. of the IADIS International Conference WWW/Internet 2007*, IADIS Press, Vila Real, Portugal, Oct-2007.
11. A. Al-Ajlan and H. Zedan, 'Towards the dynamic web services selection of e-learning services', in *Proc. of the Saudi Innovation Conference 2007*, ICT Workshop, Newcastle University, Saudi Clubs in UK and Ireland term 26th, Newcastle, UK, May-2007.

Appendix B

Glossary of Acronyms Used

ACP	Access Control Policy
AHS	Ajlan's High School
AGG	Attributed Graph Grammar
API	Application Programming Interface
AICC	Aviation Industry Computer-Based Training Committee
ATutor	Open Source Web-based Learning Content Management System
BEEP	Blocks Extensible Exchange Protocol
COBOL	COmmon Business-Oriented Language
C/S	Client/Server
CAL	Computer Assisted Learning
CMS	Course Management Systems
CVS	Concurrent Versioning System
DSs	Distributed Systems
EAI	Enterprise Application Integration
ESP	External Service Provider
FSM	Finite State Machine
FTP	File Transfer Protocol

GPL	General Public License
GUI	Graphical User Interface
GT	Graph Transformations
GXA	Global XML Web services Architecture
JFLAP	Java Formal Languages and Automata Package
J2EE	Java Platform, Enterprise Edition
JIT	just-in-time
ILIAS	open source web-based learning management system
IT	Information Technology
HTTP	Hypertext Transfer Protocol
LS	Legacy Systems
LIS	Legacy Information System
LCMS	Learning Content Management System
LON-CAPA	e-learning platform
LDAP	Lightweight Directory Access Protocol
ODBC	Open Database Connectivity
OO	Object-Oriented
OMT	Object Modeling Technique
OOSE	Object-Oriented Software Engineering
OSS	Open Source Software
OSS	Source Software
VLE	Virtual Learning Environments
Moodle	Modular Object-Oriented Dynamic Learning Environment
MLE	Managed Learning Environment
Sakai	community of academic institutions
SSiL	Saudi School in Leicester
SOAP	Simple Object Access Protocol

SCP	Social Constructionist Pedagogy
SLAs	Service Level Agreements
SLM	Service Level Management
SMTP	Simple Mail Transfer Protocol
SMS	Short Message Service
SQL	Structured Query Language)
SCP	Social Constructionist Pedagogy
SP	Service Provider
SReq	Service Requestor
SReg	Service Registry
PHP	Personal Home Page
LMS	Learning Management System
KSA	Kingdom of Saudi Arabia
SCORM	Sharable Content Object Reference Model
RSC	Rational Software Corporation
WIS	Information Systems
WSMS	Web Services Matching and Selection
WSPS	Web Services Protocol Stack
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WSM	Web Services Management
WSDL	Web Service Description Language
WIS	Web Information Systems
XML	Extensible Markup Language
URL	Uniform Resource Locator
UML	Unified Modeling Language
URI	Uniform Resource Identifier
UDDI	Universal Description, Discovery, and Integration

Appendix C

The Architecture of NuSOAP Package

The architecture of NuSOAP package has nine classes as in Figure 8.1. The main class is `nusoap_base` which has 28 methods as in Figure C.1. The Author of this package is Dietrich Ayala dietrich@ganx4.com, and Scott Nichol snichol@users.sourceforge.net and the version that we are working with it is \$Id: nusoap.php,v 1.94 2005/08/04 01:27:42 snichol Exp \$.

C.1 `nusoap_base` Class

Figure 8.1 displays the name of nine classes that NuSOAP depends on them to create a service with PHP. The main class is `nusoap_base` that has 28 methods as in Figure C.1.

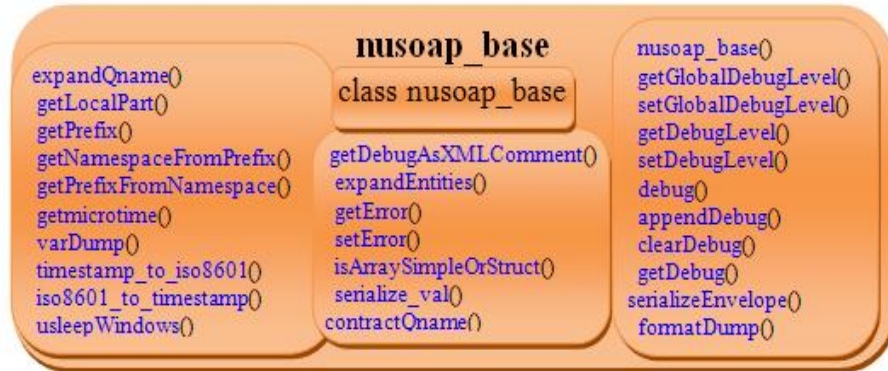


Figure C.1: The Methods in nusoap_base class

C.2 soap_fault Class

This class contains information for SOAP fault. Mainly used for returning faults from deployed functions in a server instance. This file has two classes and two methods as in Figure C.2 below.

- class nusoap_fault extends nusoap_base
- class soap_fault extends nusoap_fault



Figure C.2: The Methods in soap_fault class

C.3 nusoap_XMLschema Class

Parses an XML Schema, allows access to its data, other utility methods no validation yet, very experimental and limited. This class has two class and 16 methods as in Figure C.3.

- class nusoap_xmlschema extends nusoap_base
- class XMLSchema extends nusoap_xmlschema

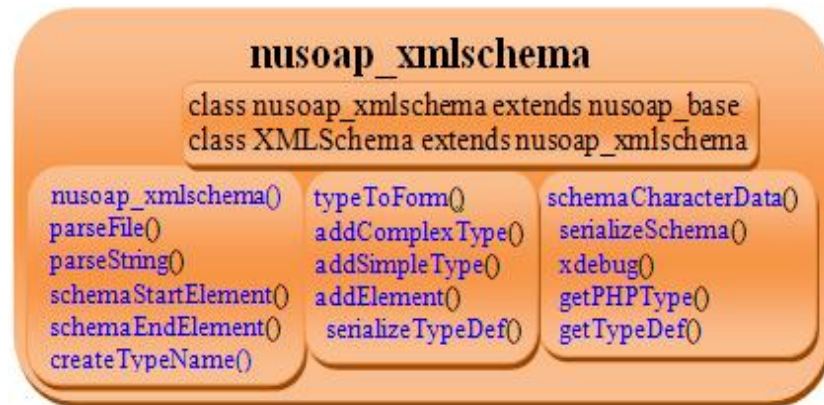


Figure C.3: The Methods in nusoap_xmlschema class

C.4 soapval Class

For creating serializable abstractions of native PHP types. This class allows element name/namespace, XSD type, and XML attributes to be associated with a value. This is extremely useful when WSDL is not used, but is also useful when WSDL is used with polymorphic types, including `xsd:anyType` and user-defined types. This class has one class and 3 methods as in Figure C.4.

- class soapval extends nusoap_base



Figure C.4: The Methods in soapval class

C.5 soap_transport_http Class

Transport class for sending/receiving data via HTTP and HTTPS. Note that PHP must be compiled with the CURL extension for HTTPS support. This class has one class and 17 methods as in Figure C.5.

- class soap_transport_http extends nusoap_base

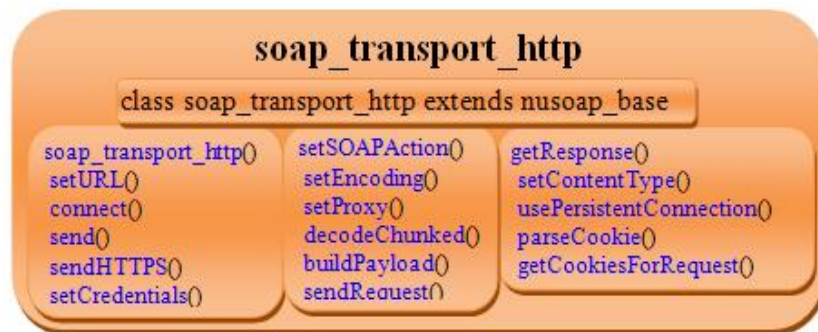


Figure C.5: The Methods in soap_transport_http class

C.6 nusoap_server Class

nusoap_server allows the user to create a SOAP server that is capable of receiving messages and returning responses. This class has one class and 17 methods as in Figure C.6 which are:

- class nusoap_server extends nusoap_base
- class soap_server extends nusoap_server

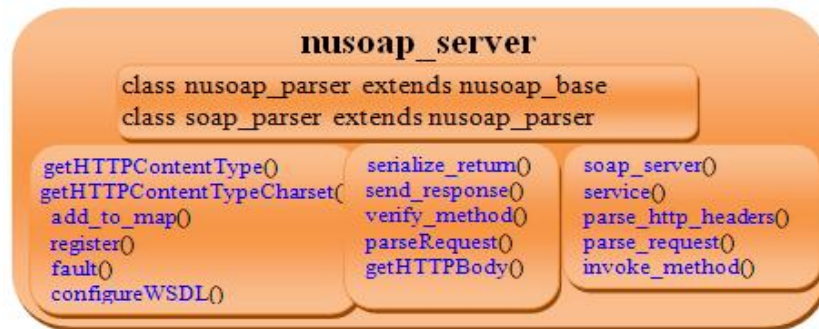


Figure C.6: The Methods in nusoap_server class

C.7 WSDL Class

Parses a WSDL file allows access to its data, other utility methods. Also builds WSDL structures programmatically. This class has one class and 26 methods as in Figure C.7.

- class wsdl extends nusoap_base

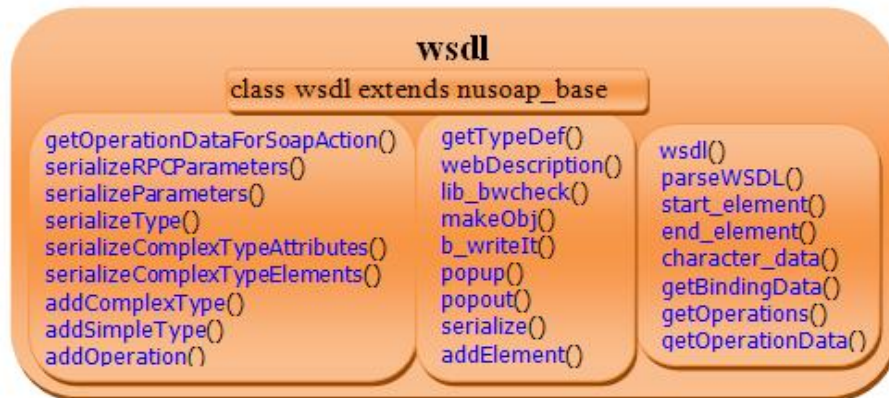


Figure C.7: The Methods in wsdl class

C.8 nusoap_parser Class

This class has two class and 8 methods as in Figure C.8 which are:

- class nusoap_parser extends nusoap_base

- class soap_parser extends nusoap_parser

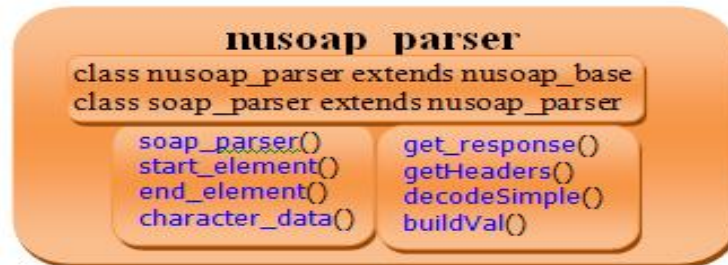


Figure C.8: The Methods in nusoap_parser Cclass

C.9 soapclient Class

soapclient higher level class for easy usage. The usage of this class is instantiate client with server info \$soapclient = new soapclient (string path [boolean wsdl]). This class has two class and 25 methods as in Figure C.9 which are:

- class nusoap_client extends nusoap_base
- lass soapclient extends nusoap_client

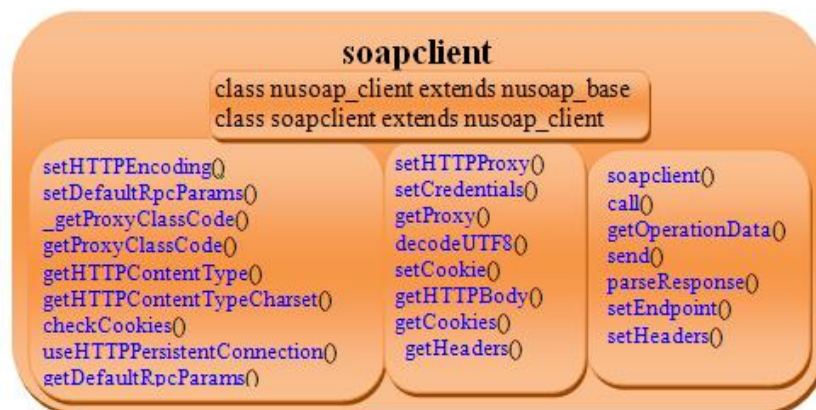


Figure C.9: The Methods in soapclient Class

Appendix D

Rules

D.1 Rule 1

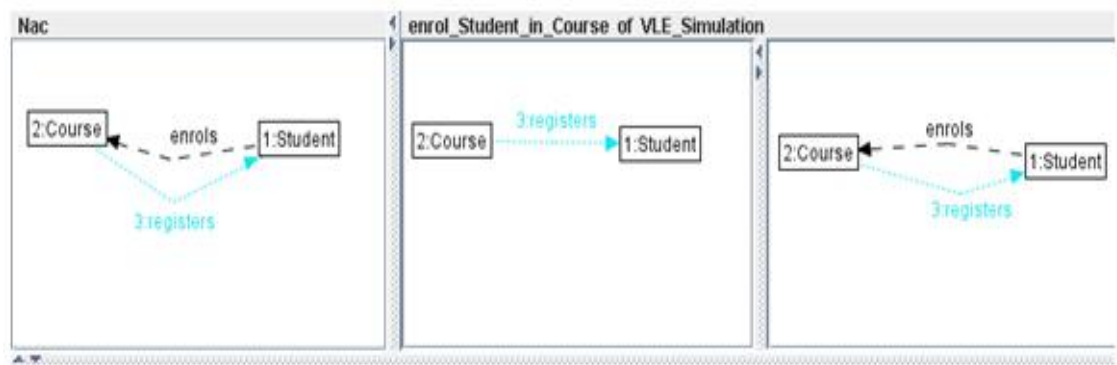


Figure D.1: Rule 1-The enrolment of a student

Rule: enrol_student.

Description: This rule says that a student cannot enrol on a course unless that student is already registered with the teacher for that course.

Parameters: Course (id, name) and student (id, name).

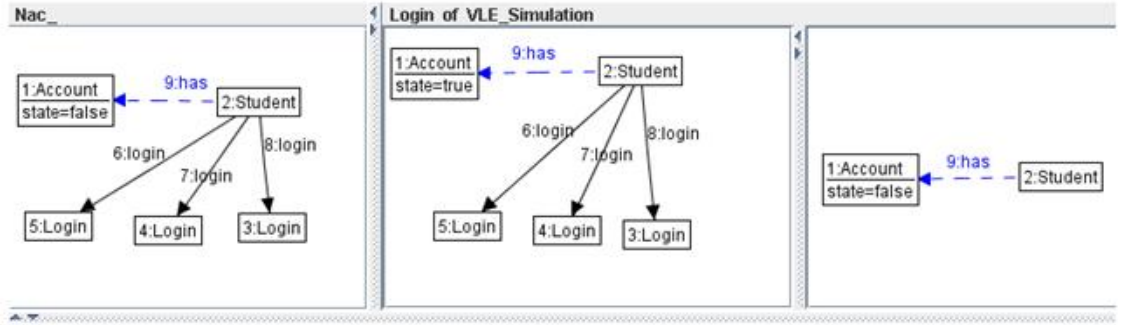


Figure D.2: Rule 2 - the login attempts of a student

D.2 Rule 2

Rule: login.

Description: This rule enables students to try three times to log in into their account, and then the account will lock.

Parameters: Account (Boolean), student (id, name) and login.

D.3 Rule 3

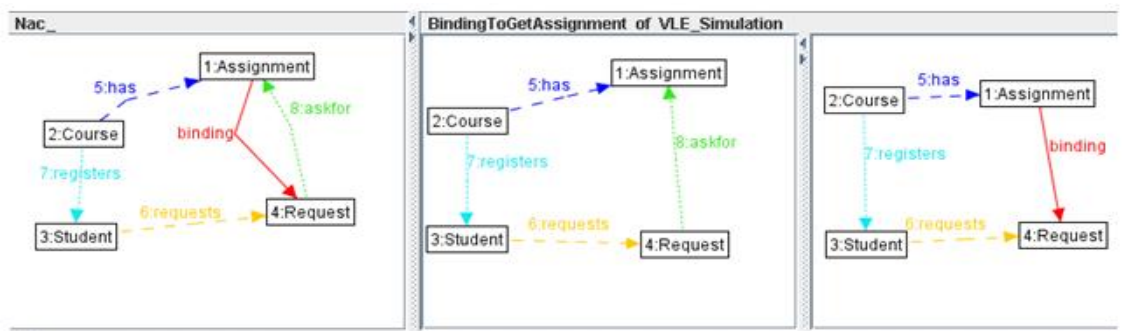


Figure D.3: Rule 3 - the binding to access the assignment service

Rule: Binding to access assignment.

Description: This rule enables students to access the assignment service only if

they have registered with the course that has this assignment, requested and asked for this assignment.

Parameters: Course (id, name), student (id, name), assignment (id, name, grade, description, assignmenType and resubmit).

D.4 Rule 4

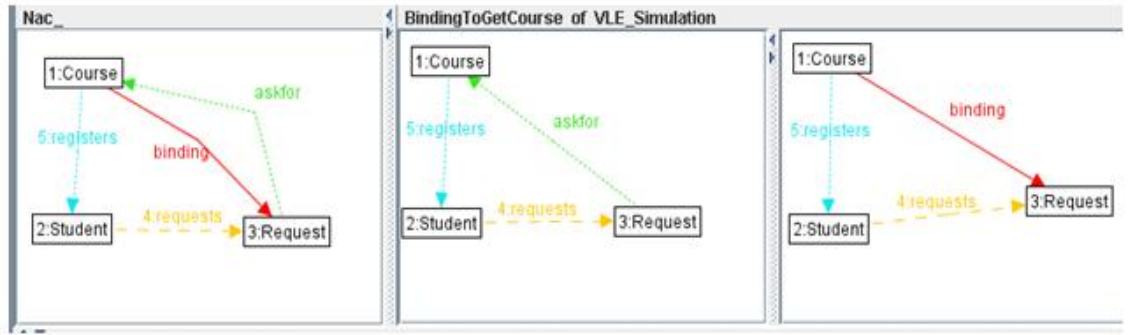


Figure D.4: Rule 4 - the binding to access the course service

Rule: Binding to access course.

Description: This rule enables students to access a course if they have already registered on the course, requested and asked for this course.

Parameters: Course (id, name), student (id, name), request.

D.5 Rule 5

Rule: managing course.

Description: This rule enables teachers to manage a course only if they responsible for this course and also have students enrolled (not an empty course).

Parameters: Teacher (id, name), course (id, name), student (id, name), request.

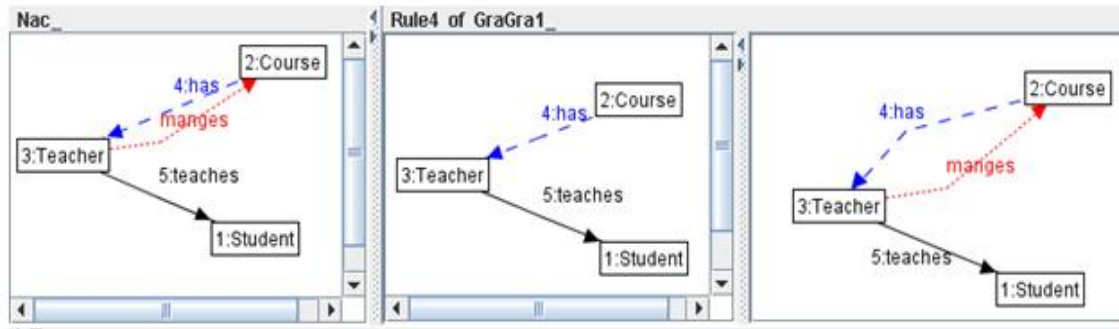


Figure D.5: Rule 5- Teacher manages course

D.6 Rule 6

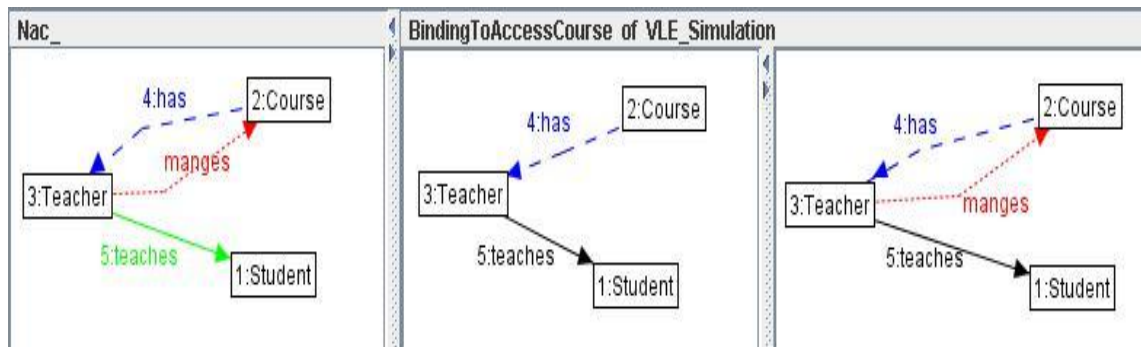


Figure D.6: Rule 6- Teacher manages course

Rule: payment.

Description: This rule enables students to pay the sees only if they have account.

Parameters: student (id, name), account (Boolean), payment (Boolean).

D.7 Rule 7

Rule: Binding to access assignment.

Description: This rule enables guest to access the assignment service only if they have requested with the course that has this assignment, requested and asked for

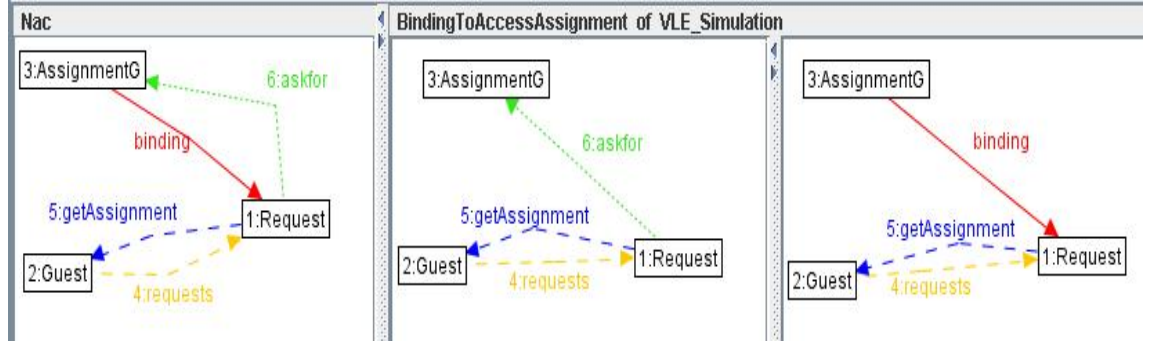


Figure D.7: Rule 7- Binding to access assignment

this assignment.

Parameters: CTeacher (id, name), course (id, name), student (id, name), request.

D.8 Rule 8

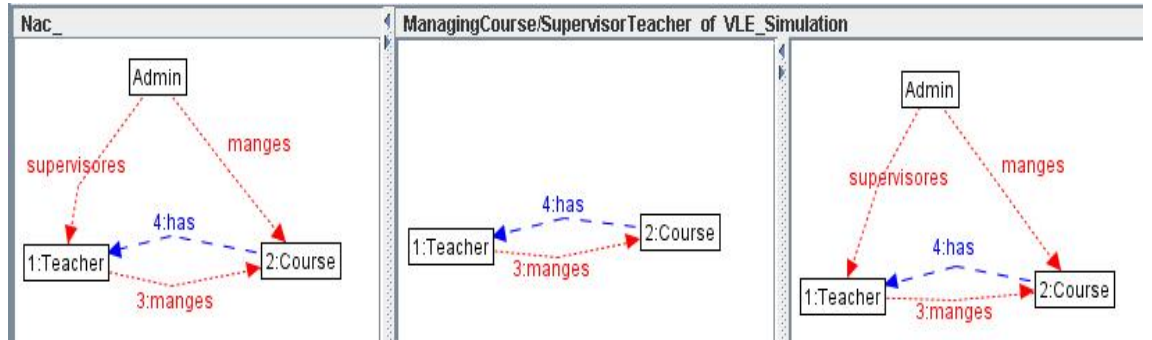


Figure D.8: Rule 8- managing course and supervisors teacher

Rule: managing course and supervisors teacher.

Description: This rule enables admin to manage a course and supervisors teacher only if they responsible for this course and also have students enrolled (not an empty course).

Parameters: Admin(id, name) Teacher (id, name), course (id, name).